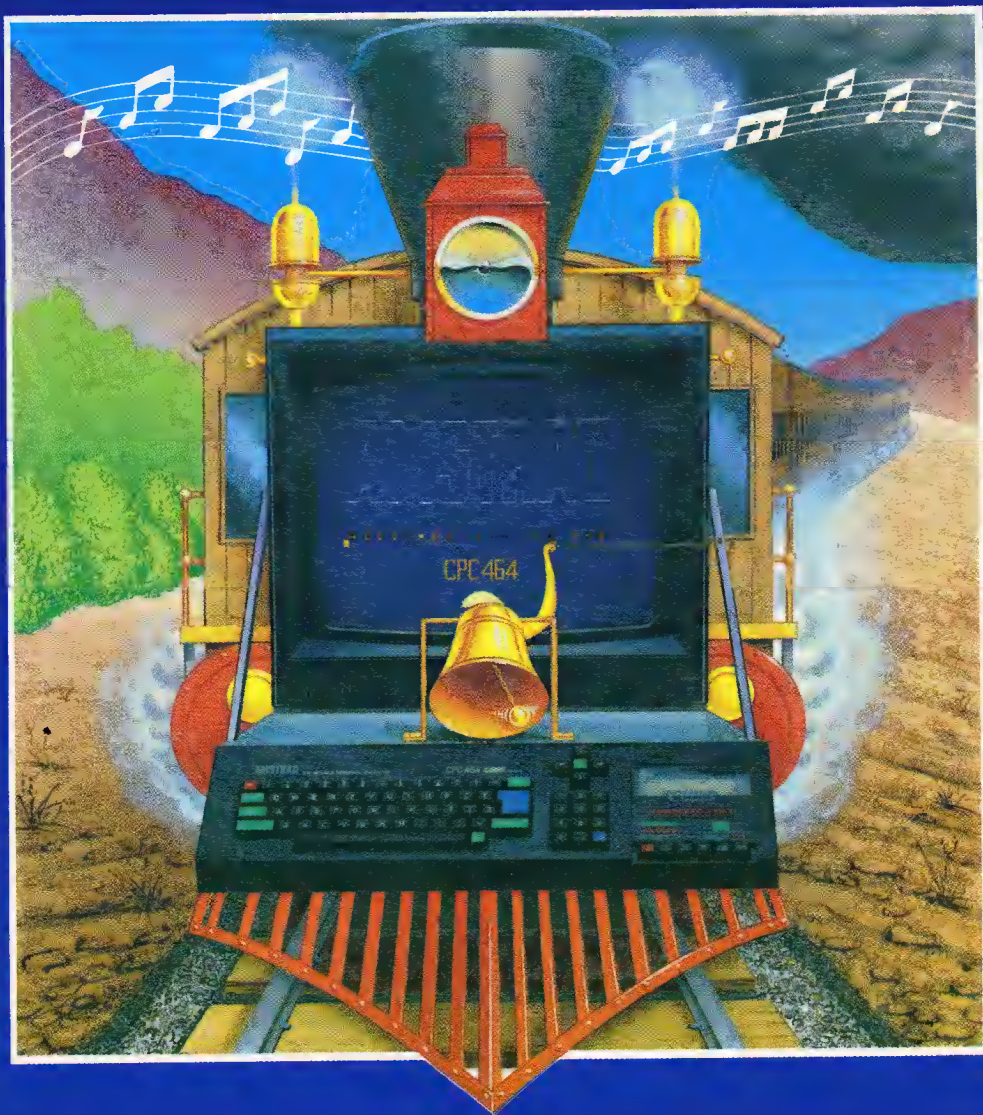


# ***Música y Sonidos con Amstrad***

**AMSTRAD**  
E S P A Ñ A

**Jeremy  
Vine**



# **Música y Sonidos con Amstrad**

*A Michael y Rosalind*

# **Música y Sonidos con Amstrad**

**Jeremy Vine**



## MUSICA Y SONIDOS CON AMSTRAD

Edición española de la obra

**BELLS AND WHISTLES ON THE AMSTRAD**

Jeremy Vine

publicada en castellano bajo licencia de

Shiva Publishing Limited  
64 Welsh Row  
Nantwich, Cheshire CW5 5ES  
Inglaterra

*Traducción:*

Emilio Benito Santos

INDESCOMP, S. A.  
Avda. del Mediterráneo, 9  
28007 Madrid

© 1984 Jeremy Vine  
© 1985 Indescomp, S. A.

Reservados todos los derechos. Prohibida la reproducción total o parcial de la obra, por cualquier medio, sin el permiso escrito de los editores.

I.S.B.N.: 28-86176-29-8  
Depósito Legal: 10948-1985

*Impresión:* Gráficas Lormo. Isabel Méndez, 15. 28038-Madrid.

*Producción de la edición española:*

Vector Ediciones  
Gutierre de Cetina, 61  
28017 Madrid

---

# Contenido

<i>Prólogo</i> .....	7
1 ¿Qué voy a aprender? .....	9
2 Introducción al sonido .....	11
3 Sonilandia .....	15
4 Interludio musical .....	25
5 El sonido de la música .....	33
6 La envolvente de volumen .....	37
7 La envolvente de tono .....	43
8 Efectos especiales .....	49
9 Música, maestro .....	55
10 Ritmo .....	59
11 Campanas y silbidos .....	61
<i>Apéndice A</i> Palabras reservadas del BASIC de Amstrad .....	63
<i>Apéndice B</i> Los parámetros de SOUND .....	69
<i>Apéndice C</i> Parámetros de la envolvente de volumen .....	71
<i>Apéndice D</i> Parámetros de la envolvente de tono .....	73
<i>Apéndice E</i> Tabla de notas, frecuencias y números de tono .....	75
<i>Apéndice F</i> El generador de sonido: notas técnicas .....	79

---

## Prólogo

El gran atractivo de los microordenadores domésticos se debe a los efectos que ejercen sobre nuestros sentidos. La vista y el oído, además de permitirnos hacer una vida normal, pueden procurarnos placer a través de aquello que vemos y oímos. En este área, los ordenadores pueden ser inigualables. Los juegos para ordenadores son tan populares gracias a los efectos visuales que incorporan, y en esto el Amstrad no se queda atrás. Pero los efectos especiales no son solamente visuales. Los ordenadores pueden generar sonidos curiosos, variados y llamativos. De esto es de lo que trata el libro.

El Amstrad CPC 464 tiene incorporado un circuito integrado generador de sonidos. Las instrucciones de BASIC que lo controlan parecen a primera vista complicadas y desconcertantes, pero trataré de explicarlas y demostrar lo fácil que es producir sonidos en el Amstrad. No doy por supuestos demasiados conocimientos por parte del lector; todos los programas han sido escritos teniendo in mente al principiante. No obstante, para aprovechar al máximo la programación de sonidos, es conveniente que el lector tenga unos conocimientos, siquiera rudimentarios, de BASIC.

Vamos a exponer los principios fundamentales de la música y el sonido y a explicar cómo se pueden programar efectos sonoros, tales como ruido de explosiones, y simular el sonido de diversos instrumentos musicales. Usted podrá también utilizar el teclado del Amstrad como si fuera el teclado de un piano. Daremos programas con los que interpretar todo tipo de música, desde simples escalas hasta jazz. Sin embargo, lo más apasionante para el lector llegará cuando sea capaz de programar los sonidos de su gusto; cuando termine de leer este libro estará en condiciones de hacerlo.

Termino con una nota personal. Éste es mi tercer libro; debo agradecer el constante apoyo y aliento de mi familia y amigos a lo largo de muchas y muy duras semanas. Mención especial merecen mis amigos del mundo de las revistas, en particular Tony Quinn, de *Acorn User*, quien publicó mis primeros trabajos y no ha dejado de ayudarme desde entonces.

Londres, 1984

Jeremy Vine

### **Sobre el autor**

Jeremy Vine estudió en la William Ellis Grammar School, Highgate, Londres, y en el City of London Polytechnic, donde se licenció en Psicología. Su primer contacto con ordenadores tuvo lugar con ocasión de sus estudios de licenciatura; desde entonces no se ha alejado de ellos. Cuando dejó el Politécnico, Jeremy trabajó por libre para la revista *Acorn User*, antes de incorporarse a Acorn Computers Limited. Durante este tiempo también estudió para un Master en Neurofisiología. Ahora, tras abandonar Acorn, dedica todo su tiempo a trabajar como escritor autónomo. Además de los libros para Shiva, publica con regularidad en varias revistas de informática, de una de las cuales es editor consultivo. Sus aficiones son tenis, piano, fotografía y, naturalmente, ordenadores.



---

## ¿Qué voy a aprender?

La compra de su Amstrad pudo haber estado motivada por su irrefrenable deseo de jugar a los marcianos o quizá porque pensaba dedicarlo a aplicaciones más serias: programas educativos, proceso de textos o incluso aplicaciones comerciales. No es probable, en cambio, que lo comprara por el generador de sonido que tiene incorporado. Quizá supiera que el Amstrad puede producir sonidos, pero seguramente no cuán variados pueden llegar a ser. Sin embargo, ahora que ha adquirido este libro, se dará cuenta de que su Amstrad puede producir bastante más que un simple «bip». De hecho, las magníficas posibilidades sonoras de este ordenador pueden añadir una nueva dimensión a todas las demás aplicaciones.

El Amstrad es una máquina versátil; su capacidad de sonido es tan sobresaliente como todas las demás. Pero, claro, no se podía esperar menos de un fabricante de alta fidelidad. ¿Se imagina a Amstrad fabricando una máquina silenciosa?

El libro tiene dos objetivos principales. En primer lugar, le enseña los rudimentos de la generación del sonido y la música. Por si usted no sabe tocar ningún instrumento musical ni leer partituras, hay una introducción a estos temas que le dará todos los conocimientos necesarios para programar música en el Amstrad.

El segundo objetivo del libro es explicarle las instrucciones que controlan los efectos sonoros. El área de la música y el sonido es muy amplia, pero las instrucciones de BASIC del Amstrad han sido diseñadas para hacer lo más sencillo posible el control de todos los aspectos de la generación de sonido.

Así pues, ¿qué va usted a aprender? La respuesta es que casi todo lo que pueda necesitar en este campo: la música de todos los tipos estará al alcance de sus dedos. Para quienes prefieran los ruidos y efectos sonoros, el Amstrad ofrece una notable capacidad para simular los sonidos que nos rodean.

En este libro damos ejemplos de timbres telefónicos, alarmas, explosiones y silbidos. Hablaremos de estos efectos y de muchos más, con el propósito de inducirle a experimentar y descubrir sus propios efectos especiales.

En todos los listados del libro las palabras reservadas de BASIC aparecen en mayúsculas, pero usted puede teclearlas en minúsculas si lo prefiere; el ordenador hará la conversión automáticamente.

Y ahora comienza la fiesta. Siéntese, relájese y encienda su Amstrad. Pronto empezará a oír, no sólo el ruido de las teclas, sino también música. ¡La aventura va a ser sonada!

---

## Introducción al sonido

La generación de sonidos en el Amstrad es realmente fácil, pero, por supuesto, es necesario conocer y entender todas las instrucciones que controlan el generador. Estas instrucciones, hay que reconocerlo, son complicadas; pero vale la pena hacer un pequeño esfuerzo.

Antes de abordar la primera instrucción necesitamos dar un repaso a los principios fundamentales del sonido. Esto le dará una base firme para saber qué efecto quiere conseguir y así poder utilizar correctamente las instrucciones adecuadas en cada caso.

### CARACTERÍSTICAS DEL SONIDO

Imagine un sonido cualquiera: un ruido estrepitoso, un suave pitido o una balada. Si es sonido, tendrá características comunes con los demás sonidos. Las características de todo sonido son: intensidad, tono y duración.

#### Intensidad

La intensidad es la característica que percibimos como «volumen»; hace que un sonido o ruido nos parezca más fuerte o más suave. En el Amstrad el volumen global de todos los sonidos generadores se controla mediante el mando de volumen situado al lado derecho de la carcasa. Pero la intensidad de cada nota se puede controlar por programa poniendo en el lugar correspondiente de la instrucción adecuada un número, del 0 al 7. Ahora bien, para definir completamente la intensidad de una nota esto no es suficiente. Las notas no sólo pueden ser más o menos intensas, sino que su intensidad puede variar mientras están sonando. No se preocupe: esto no es tan difícil como parece.

Cuando se percute una tecla del piano, la intensidad del volumen producido crece al principio (durante la fase denominada «de ataque») y luego decrece (en la fase «de declive»). Algo similar ocurre en todos los instrumentos; las diferencias están en que el ataque y el declive sean más o menos rápidos. Tampoco esto debe preocuparle por el momento. En el capítulo 6 estudiaremos las envolventes de volumen, que son las responsables de estas variaciones de intensidad.

Bueno; la amplitud del sonido ha resultado una característica no tan sencilla. Y no es eso todo. La sensación de volumen depende también de otros factores; por ejemplo, de la duración. En efecto, un sonido que dure solamente una fracción de segundo parecerá más suave que uno que dure más tiempo, aunque ambos tengan en realidad la misma intensidad.

## Tono

Dicho con palabras sencillas, el tono es la característica que hace que los sonidos den la sensación de ser más o menos graves o agudos. Más técnicamente, lo que caracteriza el tono es la frecuencia, esto es, el número de ondas de sonido por segundo. Los humanos solamente podemos oír los sonidos cuya frecuencia esté dentro de ciertos márgenes. Un ejemplo clásico es el del silbato que pueden oír los perros pero no los hombres. La razón es que los perros pueden percibir frecuencias más altas que las personas. El hecho de que nosotros no podamos oír un sonido no significa que éste no exista.

La frecuencia, para nuestros fines, está relacionada con las escalas musicales. Las frecuencias más altas corresponden a notas más agudas. En el piano cada nota tiene una frecuencia, pero en el ordenador tenemos que especificar la frecuencia de la nota que deseamos generar. Más adelante nos encontraremos la instrucción ENT, que controla la envolvente de tono. Mediante ella podremos especificar la forma de variación del tono a lo largo del tiempo en que está sonando una nota. ¿Para qué? Veamos: el lector habrá observado que en algunos instrumentos el músico puede variar el tono de una nota por medio de lo que denominamos *vibrato*. La instrucción ENT nos permitirá simular ese efecto. Volveremos a hablar del tono cuando tengamos que incluirlo en las instrucciones que generan los sonidos.

## Duración

La duración de una nota es, como el lector seguramente ha adivinado, el tiempo durante el que ésta está sonando. En todas las composiciones mu-

sicales se especifica la duración que ha de tener cada una de las notas (véase el capítulo 4) y la velocidad a la que la pieza ha de ser ejecutada. Estas variaciones de la duración de las notas son, entre otras, las que producen los diferentes ritmos que conocemos, desde el vals hasta el rock and roll.

Las instrucciones de sonidos del Amstrad permiten especificar la duración de las notas.

Con todos estos recursos disponemos los medios necesarios para programar una gran variedad de sonidos. Pero las posibilidades de control del generador del sonido del Amstrad son aún más amplias.

### Canales de sonido

Hasta ahora solamente hemos hablado de notas aisladas o de series de notas que se suceden en el tiempo. Sin embargo, cuando escuchamos a un buen pianista lo que escuchamos no son notas aisladas. El pianista muchas veces hace sonar varias notas a un tiempo, sincronizadas, para formar acordes. También esto lo podemos simular en el Amstrad.

Este ordenador tiene *tres* canales de sonido. El programador puede elegir que los canales suenen al unísono o por separado. Los canales llevan los nombres de A, B y C. Como veremos más adelante, no sólo podemos utilizarlos para ejecutar tres voces sincronizadas, sino también para reproducir tres sonidos distintos al mismo tiempo.

Aparte de las notas musicales, el Amstrad puede generar otros sonidos, los llamados efectos «de ruido blanco». En ellos se basan los ruidos de explosiones y otros similares que podemos escuchar en muchos juegos, así como los efectos de percusión que podemos combinar con la música.

El ruido blanco es sencillamente ruido aleatorio, esto es, una mezcla equilibrada de tonos de todas las frecuencias posibles. El ruido, en este ordenador, sólo se puede enviar a uno de los tres canales. O sea, mientras que se pueden reproducir tres notas distintas al mismo tiempo, el ruido sólo puede sonar por un canal en un momento dado.

El Amstrad almacena las instrucciones de sonido en «colas». Cada canal tiene su propia cola, y en cada una hay espacio para un máximo de cinco instrucciones. Una de ellas será la activa (la que está sonando en el momento) y las otras estarán esperando. Cada vez que termina la ejecución de una instrucción, el ordenador envía a la cola una instrucción nueva. Esto hace que el generador de sonido sea relativamente autónomo, con lo que el resto del ordenador puede atender a otras tareas sin tener que esperar hasta que haya terminado de sonar el sonido programado.

## **¿SUENA COMPLICADO?**

Todo esto puede parecerle elemental, o quizá extremadamente complejo. Cualquiera que sea su base en música y programación, este libro no le dejará abandonado. De todo lo dicho volveremos a ocuparnos cuando llegue el momento, y entonces daremos todas las explicaciones necesarias.

Pero empecemos ya a estudiar las instrucciones de BASIC con las que se programan los sonidos del Amstrad.

---

## Sonilandia

Bien, hemos hablado de todas esas magníficas posibilidades sonoras del Amstrad CPC 464, pero ¿qué tenemos que hacer para que el altavoz empiece a sonar? Lo primero, subir el mando de volumen al máximo y dejarlo así (si la familia no se opone). Ahora teclee la siguiente orden (y no olvide pulsar ENTER al final):

**SOUND 1,478**

La nota que ha oído es DO media, con duración de un cincuentavo de segundo. La instrucción SOUND va seguida de varios parámetros, siete como máximo, con los que se controlan todas las características del sonido. Si está impaciente por saber para qué sirven, vea la figura 3.1.







De momento sólo hemos utilizado dos de ellos, los únicos imprescindibles. Su forma general es:

**SOUND C,T**

Vamos a estudiarlo con detalle.

### EL NÚMERO DE SELECCIÓN DE CANALES

El primer parámetro de SOUND, representado por C, especifica qué canal (o canales) debe sonar y en qué condiciones. Por ahora no vamos a ocuparnos de esas condiciones; nos limitaremos a averiguar cómo seleccionar canales. En el ejemplo anterior C tenía el valor 1, lo que significa que seleccionábamos el canal A. En el capítulo 2 hemos dicho que hay tres canales y sin embargo el margen de valores de C se extiende de 1 a 255. Por el momento sólo nos interesan los primeros números, con los que, según se indica en la tabla 3.1, se seleccionan los tres canales individualmente. Así,

Canal	Valor
 <b>A</b> 	<b>1</b>
 <b>B</b> 	<b>2</b>
 <b>C</b> 	<b>4</b>

**Tabla 3.1** Números de selección de canales: acceso a los tres canales de sonido.

la instrucción **SOUND 2** selecciona el canal **B**. Pero no teclee sólo **SOUND 2** porque, como recordará, esta instrucción requiere al menos dos parámetros. Más adelante veremos qué efecto tienen los otros valores posibles de **C**.

## EL NÚMERO DE TONO

El segundo parámetro, **T**, controla el tono del sonido. En el ejemplo anterior pusimos el número 478, que corresponde a la nota **DO** media. ¿Cómo podemos saber qué número hay que utilizar para obtener una nota determinada? Muy sencillo: como el lector sabe, cada nota tiene una frecuencia, y a cada frecuencia corresponde un número de tono; en el apéndice E se da una relación de las notas, las frecuencias y los números de tono correspondientes. El parámetro **T** tiene que estar comprendido entre 0 y 4095. El valor que se utilice en cada caso puede corresponder a una nota determinada (por ejemplo, el número 379 produce la nota **MI**), pero también puede ser un valor intermedio entre notas, cualquiera dentro del margen mencionado. Tenga en cuenta que el número que ponga como segundo parámetro de **SOUND** tiene que ser el número de tono, no la frecuencia. Como puede ver, cada frecuencia tiene un número de tono asociado.

Cambie el valor de **T** para observar cómo varía el tono. Teclee, por ejemplo, las siguientes órdenes:



SOUND 1,956  
 SOUND 1,478  
 SOUND 1,60

Todas ellas producen la nota DO, pero de diferentes escalas. Quizá haya observado algo que puede parecerle extraño: cuanto menor es el número de tono, más aguda es la nota. Este hecho es característico del Amstrad; no lo olvide si quiere evitar confusiones.

¿Qué ocurre si como número de tono ponemos el 0? No habremos especificado ninguna frecuencia, lo que nos servirá para crear efectos de «ruido blanco».

## EL NÚMERO DE DURACIÓN

Pasemos al siguiente parámetro, D, que especifica la duración del sonido, es decir, durante cuánto tiempo va a estar sonando la nota. Teclee la siguiente orden:

SOUND 1,568,100

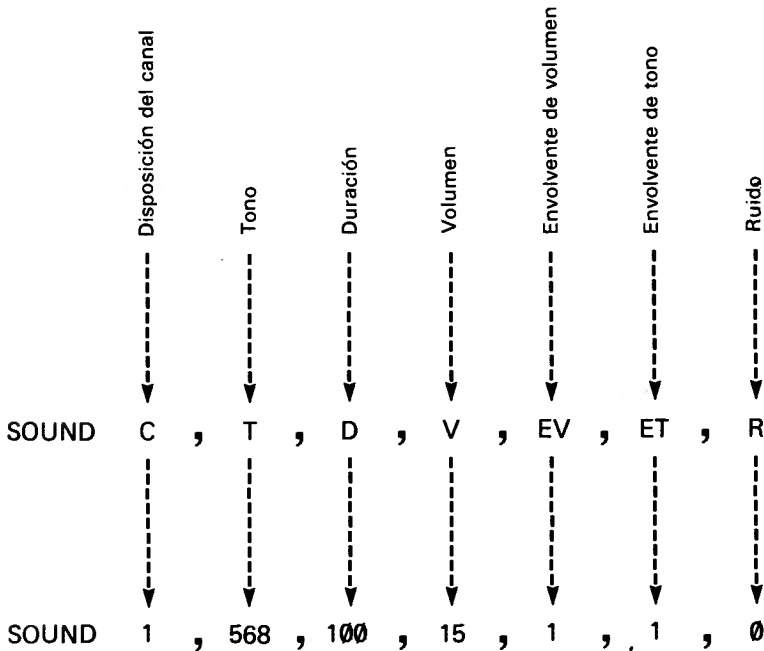


Fig. 3.1 Los parámetros de SOUND.

Acaba de oír la nota LA por el canal A durante un segundo. El tercer parámetro especifica la duración en unidades de centésimas de segundo; o sea, D=100 especifica un segundo. Toda duración mayor que 0 será obedecida como tal. Sin embargo, el número de duración nos va a servir para otras cosas.

En primer lugar, cuando D es cero, le estamos diciendo al Amstrad que la duración va a ser controlada por la envolvente de volumen asociada al canal. No está claro, ¿verdad? Lo explicaremos cuando lleguemos a la instrucción ENV.

El otro tipo «raro» de número de duración es un número negativo. También hace referencia a la envolvente de volumen; concretamente, indica al ordenador que debe repetir esa envolvente el número de veces especificado. Por ejemplo, el número de duración -3 hará que la envolvente suene 3 veces.

Experimente con diversos valores de D para hacerse una idea del efecto de la duración de las notas. En la tabla 3.2 se da un resumen de la información relativa al número de duración.

Valor	Efecto
Implícito: 20	—
Margen: -32768 a +32767	—
> 0	Duración del sonido en centésimas de segundo
= 0	Duración controlada por la envolvente de volumen (ENV)
< 0	Número de veces que se repite el sonido descrito por la envolvente de volumen

**Tabla 3.2** Efectos del valor del número de duración, D.

Si no se especifica número de duración, como ocurría en el primer ejemplo de este capítulo, el ordenador toma el valor implícito, que es el 20, lo que da notas de un quinto de segundo.

## EL NÚMERO DE VOLUMEN

Veamos con el cuarto parámetro, V, que especifica el volumen. Teclee el programa 3.1 y ejecútelo.

### Programa 3.1

```
10 FOR volumen=0 TO 7
20 SOUND 1,478,25,volumen
30 NEXT
```

El volumen va aumentando con cada pasada por el bucle; cuando el número llega a 7, el sonido es de la máxima intensidad posible. El margen de valores es, pues, de 0 a 7. El número 0 especifica intensidad nula.

Si se ha especificado envolvente de volumen ( $D=0$ ), el margen del número de volumen va de 0 a 15; en este caso el número no especifica ningún volumen, sino que hace referencia a la envolvente de volumen que se debe utilizar.

Estos cuatro parámetros son los fundamentales de la instrucción SOUND; sólo con ellos ya es posible programar buena música. En el capítulo 5 veremos cómo hacerlo. No obstante, todavía hay otros tres parámetros más. Los dos que vienen ahora tienen que ver con las instrucciones de envolvente de volumen (ENV) y de tono (ENT), que explicaremos en los capítulos 6 y 7, respectivamente. Por ahora nos limitaremos a decir cómo encajan en la instrucción SOUND.

## **ENVOLVENTES DEFINIDAS POR EL USUARIO: ENV y ENT**

El quinto parámetro EV, se incluye en la instrucción SOUND cuando se ha especificado una envolvente de volumen, la cual, como el lector ya sabe, describe la variación del volumen de una nota con el tiempo.

El valor de EV debe estar en el margen de 0 a 15. El valor 0 especifica que no se debe utilizar ninguna envolvente. Los valores positivos especifican el número de envolvente que se ha de utilizar. Por ejemplo, si  $EV=4$ , el sonido estará controlado por la envolvente número 4.

El sexto parámetro, ET, funciona igual que EV, pero referido a las envolventes de tono. El margen de valores va también de 0 a 15.

Cuando se utilizan estas opciones se empieza a aprovechar los refinamientos de este sorprendente sistema de sonido; de hecho, son la base de muchos efectos sonoros que crearemos más adelante.

Finalmente...

## **EL NÚMERO DE RUIDO**

Ya hemos dicho antes que con el ordenador Amstrad podemos crear efectos de «ruido blanco». El último parámetro, R, es el «número de todo del ruido». Especifica si se ha de producir ruido y, si es así, de qué frecuencia. Cuando se omite este parámetro, o cuando su valor es cero, no se genera ruido. El margen utilizable es de 0 a 31. Para observar el efecto de este parámetro, teclee y ejecute el programa 3.2.

**Programa 3.2**

```

10 ENV 15,43,-68,3
20 FOR volumen=0 TO 7
30 SOUND 1,478,25,volumen,15,0,0
40 NEXT

```

Si así lo ha hecho, habrá oído unos ruidos cortos e intermitentes. Observe que hemos utilizado todos los parámetros de SOUND. Al parámetro EV le hemos dado el valor 15 porque queríamos usar la envolvente número 15, definida en la línea 10. No trate de descifrar la instrucción ENV; ya la explicaremos cuando llegue el momento. El sexto parámetro, ET, es cero porque todavía no hemos definido ninguna envolvente de tono. El último parámetro es cero, por lo que no se produce ruido. Pero podemos modificarlo; cambie la línea 30 para que sea

```
30 SOUND 1,478,25,volumen,15,0,10
```

donde, como puede observar, no hemos hecho más que cambiar el séptimo parámetro. ¿Nota el efecto? Si quiere apreciar la diferencia entre un tono puro y un ruido teclee las dos órdenes siguientes:

```

SOUND 1,478,100,15
SOUND 1,478,100,10,0,0,10

```

Con esto hemos terminado la descripción de los parámetros de la instrucción SOUND. Usted queda así equipado con las herramientas necesarias para empezar a jugar con esta instrucción. Puede parecer un poco complicada al principio; para dominarla, teclee unas cuantas instrucciones y vaya cambiando los valores de los parámetros para observar los efectos. Para facilitarle esta labor hemos incluido el programa 3.3. Ejecútelo cuantas veces desee; cuando esté seguro de que ha comprendido a la perfección el funcionamiento de esta instrucción, pase al siguiente apartado.

**Programa 3.3**

```

10 CLS
20 PRINT"Elija numero de tono (0-4095)"
30 INPUT t
40 PRINT"Elija numero de duracion (1-32767)"
50 INPUT d
60 PRINT"Elija volumen (0-7)"

```

```

70 INPUT v
80 PRINT"Elija ruido (0-31)"
90 INPUT r
100 SOUND 1,t,d,v,0,0,r
110 PRINT"Pulse una tecla para continuar"
120 x$=INKEY$:IF x$="" THEN 120 ELSE 10

```

## CÓCTEL DE CANALES

Al empezar este capítulo describimos el primer parámetro de SOUND: el número de selección de canales. Como dijimos entonces, este número sirve para más cosas que la mera selección de uno de los tres canales. Antes de nada, veamos la tabla 3.3, que es una versión más completa de la tabla 3.1. La disposición de los canales indica a la máquina qué canal debe sonar, si debe hacerlo al unísono con otro canal, si el sonido del canal se retiene o si se lo borra de la cola de sonidos.

Los valores que se muestran en la tabla 3.3 envían al generador de sonido las órdenes pertinentes. Sumando varios de estos valores podemos combinar diversos efectos. Por ejemplo, si queremos enviar sonido a los canales A y B escribimos lo siguiente:

```
SOUND 3,478,50,7
```

porque 1 (canal A)+2 (canal B)=3.

Para poder apreciar el efecto de la selección de varios canales, teclee y ejecute el siguiente programa.

### Programa 3.4

```

10 KEY 128,"SOUND 1,478,50,7"+CHR$(13)
20 KEY 138,"SOUND 3,478,50,7"+CHR$(13)
30 KEY 138,"SOUND 5,478,50,7"+CHR$(13)
40 KEY 138,"SOUND 7,478,50,7"+CHR$(13)

```

Hemos utilizado la instrucción KEY para definir cuatro teclas. Cuando ejecute el programa no observará ningún efecto inmediato. Sin embargo, en cuanto vea aparecer el mensaje «Ready», las teclas habrán quedado definidas; para ejecutar las instrucciones de sonido que puede ver en el programa 3.4 tiene que pulsar CTRL al mismo tiempo que alguna de las teclas recién definidas del teclado numérico. Pruébelas en el siguiente orden:






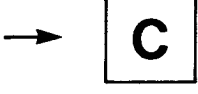
Valor	Canal(es)	Efecto
1		Seleccionar canal A
2		Seleccionar canal B
4		Seleccionar canal C
8		Sincronizar con canal A
16		Sincronizar con canal B
32		Sincronizar con canal C
64		Retener sonido
128		Borrar cola de sonidos

Tabla 3.3 Números de selección de canales (2): disposición de los canales.

CTRL+0  
CTRL+.  
CTRL+ENTER  
CTRL+3

Si escucha atentamente apreciará las diferencias entre los cuatro sonidos. De forma análoga, combinando otros valores de los números de selección de canales se pueden conseguir otros efectos. Por ejemplo, para enviar sonido al canal A en sincronismo con el B y retenerlo, el parámetro será 81 ya que

$$1 (\text{canal A}) + 16 (\text{sincronizar con el B}) + 64 (\text{mantener}) = 81$$

Estas posibilidades de elección son importantes, pues permiten al programador sincronizar canales y borrar colas cuando le convenga.

## SQ Y RELEASE

Quedan por estudiar dos instrucciones del sonido. La primera, SQ, es una función que da el número de plazas libres que quedan en la cola del sonido del canal especificado. Esto es útil también para determinar si un determinado canal está aún activo. Veamos el ejemplo del programa 3.5.

### Programa 3.5

```
10 PRINT"Esta sonando el canal A"
20 SOUND 1,261,250,7
30 PRINT"FIN"
```

Si ejecuta este programa tal como está, verá aparecer la palabra FIN inmediatamente, antes de que deje de sonar el canal A. Para determinar si el canal está activo, introducimos SQ. Añada al programa la siguiente línea:

```
25 WHILE SQ(1)>127:WEND
```

Al ejecutar la nueva versión del programa, el mensaje final no aparece mientras no haya concluido el sonido. Esto ocurre porque SQ(1) da un valor mayor que 127 mientras esté sonando el canal A. La línea 25 se repite mientras (WHILE) se cumpla la condición. El número que se pone entre paréntesis a continuación de SQ hace referencia al canal de interés.

La instrucción RELEASE libera el sonido de un canal si este sonido está retenido. Los parámetros de esta instrucción están en el margen de 0 a 7. Por ejemplo, RELEASE 1 libera el sonido que esté retenido en el canal A, si lo hay.

Con esto termina nuestro repaso de las instrucciones de sonido del Amstrad. La mejor forma de entender y llegar a dominar estas instrucciones es experimentar con la instrucción SOUND. Esto es de por sí una buena diversión, y puede proporcionar resultados inesperados. Volveremos a encontrarnos la instrucción SOUND en el capítulo 5.



---

## Interludio musical

En este capítulo vamos a dar una introducción a la terminología y la notación de la música. Si el lector conoce la notación musical, o si toca algún instrumento, dé un rápido repaso a este capítulo y pase al siguiente. Pero, dado que este libro va dirigido a todo tipo de usuarios, no quedaría completo si no explicásemos algo de la jerga de los músicos. Este capítulo no es, ni puede serlo, un curso completo; pero si lo lee detenidamente, le servirá para entender los capítulos posteriores, en los que hablaremos de cómo convertir una pieza musical en un programa de ordenador.

### PENTAGRAMAS Y NOTAS

Antes de nada, unas palabras de ánimo. No se deje abrumar por la notación musical. Para quien no sepa nada de música puede parecer muy críptica: todas esas rayas, puntos, signos extraños... Pero no es tan difícil como a usted le parece. Vamos a ver primero de qué medios se valen los compositores para poner su obra en papel. La música se escribe en papel rayado; esto no le parecerá tan raro si recuerda los cuadernos de caligrafía. En el caso de la música, las rayas están agrupadas de cinco en cinco, según se muestra en la figura 4.1. Un grupo de rayas es lo que se llama *pentagrama*.



**Fig. 4.1** Un pentagrama vacío.

Los tonos musicales se representan en el pentagrama mediante signos llamados *notas*. La posición en vertical (la altura) de una nota indica su tono.

Las notas se pueden escribir sobre las líneas o entre dos de ellas. La altura a la que se encuentra una nota indica su nombre. En la figura 4.2 damos un ejemplo. En esta figura aparecen tanto los nombres españoles de las notas (DO, RE, MI...) como los ingleses (C, D, E...).

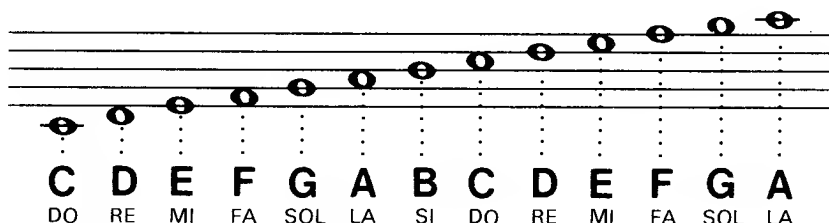


Fig. 4.2 Serie de notas en un pentagrama.

Observe que a cada siete notas los nombres se repiten; es decir, las siete primeras notas son DO, RE, MI, FA, SOL, LA, SI, y la siguiente vuelve a ser la DO, pero una octava más arriba que la primera. Si hay notas demasiado altas o demasiado bajas que no quepan en el pentagrama, su altura se indica con unas pequeñas rayas adicionales, con las que, por consiguiente, el músico puede ampliar el pentagrama. Sin embargo, no es costumbre poner demasiadas de estas rayas adicionales, pues si así se hiciera la música resultaría prácticamente ilegible.

## DURACIÓN

Ya sabemos cómo se indica el tono de las notas en las partituras. Veamos ahora cómo se representa su duración. Hay diversos signos que caracterizan las diferentes duraciones. No olvide que la duración de las notas no es lo mismo que el tempo, o velocidad a la que se las interpreta; de esto nos ocuparemos algo más tarde. Los valores de tiempo de las notas indican su duración por comparación con otras. En la tabla 4.1 se da la lista de los símbolos y duraciones.

La *redonda* es (normalmente) la nota más larga; la *fusa*, la más corta. Si se escribe un punto (.) a continuación del símbolo de una nota, la duración de ésta se multiplica por 1.5. También se pueden indicar periodos de silencio insertando pausas entre notas. En la figura 4.3 se dan los símbolos de los silencios; las duraciones son iguales que en las notas de su mismo nombre.

Si el silencio ha de ser muy largo, se pone un número sobre el signo. Por ejemplo, el número 12 puesto sobre un signo de silencio indica una pausa de 12 veces la correspondiente a ese signo.







<i>Nota</i>	<i>Nombre</i>
	Redonda
	Blanca
	Negra
	Corchea
	Semicorchea
	Fusa

Tabla 4.1 Valores de tiempo.

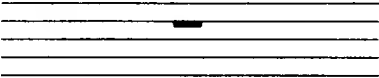
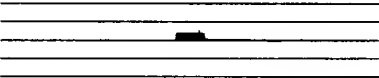

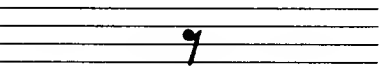
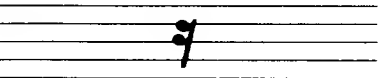

	Silencio de redonda
	Silencio de blanca
	Silencio de negra
	Silencio de corchea
	Silencio de semicorchea
	Silencio de fusa

Fig. 4.3 Silencios.

## CLAVES Y ESCALAS

Hemos dicho antes que el pentagrama se puede ampliar con unas líneas adicionales. Para facilitar la lectura de la música se pueden utilizar dos pentagramas paralelos. Habrá, pues, dos pentagramas distintos cuyo intervalo tonal se indica mediante un signo: la clave. Este signo se pone al principio del pentagrama y fija el tono de una determinada nota. Las dos claves más frecuentes, sobre todo en música para piano, son las de SOL y FA. La clave de SOL es la de tono más alto de las dos. Estas escalas están representadas en la figura 4.4.

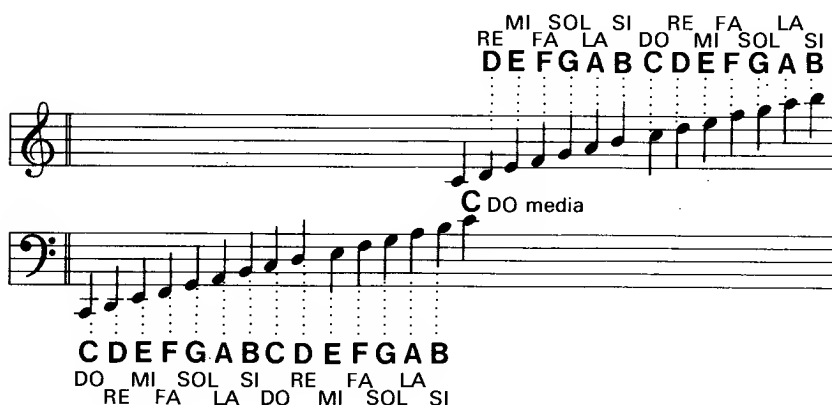


Fig. 4.4 Claves básicas.

Observe que la posición de una nota determinada depende de la clave en la que esté. La nota DO media queda entre los dos pentagramas. Cada serie de ocho notas forma una *escala*. La más sencilla es la de DO mayor. Su situación en el piano se ilustra en la figura 4.5.

Como puede ver en la figura, la escala de DO mayor sólo utiliza teclas blancas. Además, no hay teclas negras entre MI y FA ni entre SI y DO. Las notas MI y FA están muy próximas entre sí: sólo hay un semitono entre ellas. Lo mismo ocurre con SI y DO. Todos los demás pares de teclas tienen una tecla negra en medio; la distancia entre las correspondientes notas es de dos semitonos. Todas las escalas mayores responden al mismo esquema: tono-tono-semitono-tono-tono-tono-semitono. Si usted va tocando todas las teclas del piano (incluidas las negras) de izquierda a derecha, estará avanzando de semitono en semitono; cuando haya tocado doce teclas, habrá vuelto a la primera nota. Esta sucesión de 12 semitonos es una escala *cromática*.

Para ejecutar otras escalas mayores, distintas de la de DO, son necesarias

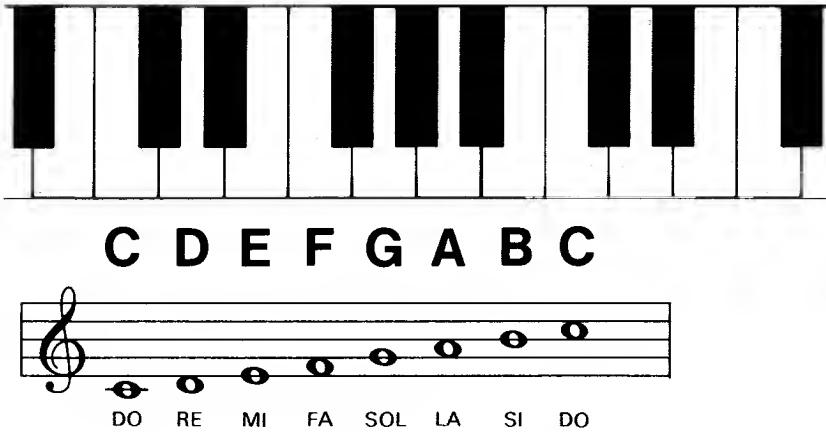


Fig. 4.5 Relación entre las notas del pentagrama y las teclas del piano.

las teclas negras. A estas teclas se las designa por el nombre de una de las dos blancas adyacentes. Por ejemplo, la que está a un semitono por encima de DO se llama DO sostenido (#); la que está por debajo, DO bemol (b). Si se ponen estos signos al principio de la partitura, se les llama *signatura de clave*. Por ejemplo, la *signatura de clave* de SOL mayor tiene un sostenido (#), que es FA#. Esto significa que cuando se interpreta la escala, la nota FA se eleva siempre un semitono. En la figura 4.6 se dan unos ejemplos de *signaturas de clave*.

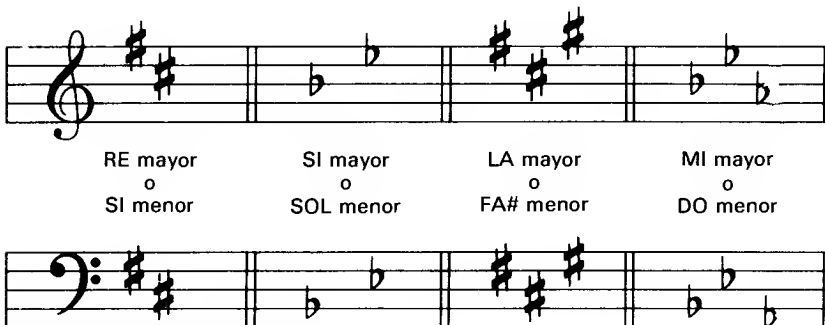
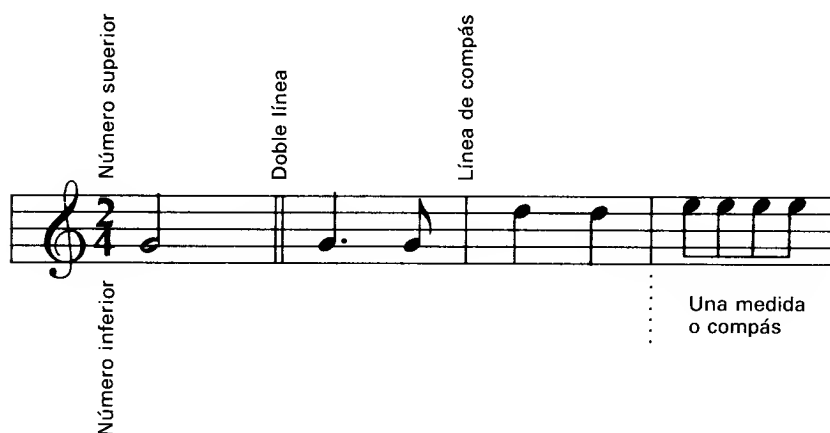


Fig. 4.6 Algunas *signaturas de clave*.

Una nota que ha sido afectada por un sostenido o un bemol puede ser devuelta a su tono original mediante el signo  $\natural$  (becuadro), que cancela el sostenido o el bemol.

## COMPASES Y RITMO

La música está dividida en porciones llamadas *medidas* o *compases*. En el pentagrama estas divisiones se indican con unas rayas verticales, las llamadas líneas de compás. Una raya vertical doble indica el final de un pasaje o pieza. Al principio de la partitura se especifica la duración de los compases mediante la signatura de tiempo. Está formada por un número superior y otro inferior; el superior indica el número de movimientos o golpes de que consta el compás; el inferior, la duración de cada uno de ellos. Véase el ejemplo de la figura 4.7.



**Fig. 4.7** Compases y valores de tiempo.

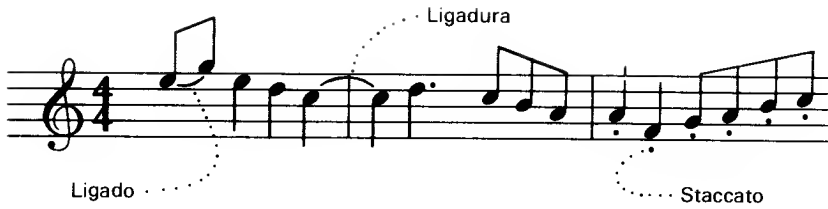
En este caso la signatura de tiempo es 2/4. Esto le indica al músico que cada compás consta de 2 golpes, y que cada golpe tiene la duración de una negra, representada por el número 4. Mediante estas notaciones se especifica el ritmo.

Otros detalles interesantes que podemos señalar son las diferentes formas en que se puede ejecutar una nota. En primer lugar, si hay un punto por encima o por debajo de una nota, esto indica que la nota se debe interpretar en «staccato», es decir, recortando la nota para dejar una breve pausa entre ella y la siguiente.

El efecto contrario es el «ligado». Si dos notas están unidas mediante un arco, las notas se deben enlazar de forma suave y continua.

Finalmente, si el arco une dos notas del mismo tono, el efecto es una «ligadura». Esto significa que se debe ejecutar una sola nota de duración igual a la suma de las de las dos notas unidas.

En la figura 4.8 se dan ejemplos de estas tres posibilidades.



**Fig. 4.8** Ligado, ligadura y staccato.

Es posible que el lector se haya perdido un poco en todas estas explicaciones. No debe preocuparse por ello, porque tampoco es necesario que se sepa toda esta terminología de memoria; le servirá, eso sí, cuando más adelante volvamos a mencionar alguno de estos puntos. En un libro de este tamaño no es posible dar un repaso completo a los fundamentos de la música, ni siquiera a nivel elemental. Si desea profundizar en la teoría musical, consulte alguno de los muchos libros que hay dedicados al tema.

A medida que vayamos avanzando, comprobará que lo que hemos explicado en este capítulo le va a servir para comprender mejor los programas musicales y para convertir las partituras a una forma inteligible para el Amstrad. Vuelva a leer este capítulo si lo cree necesario, y pase luego al capítulo 5, en el que empezaremos a escribir música en el Amstrad.





---

## El sonido de la música

La conversión de la música a una forma que el Amstrad pueda comprender es una tarea relativamente sencilla. En el capítulo 3 decíamos que el número de tono de la instrucción SOUND podía producir una nota reconocible (véase el apéndice E). En este capítulo vamos a utilizar los valores de T que hacen que el generador de sonido emita tonos musicales.

### ESCALAS CROMÁTICAS

Un buen punto de partida para entender la relación entre los números de tono del Amstrad y las notas musicales es considerar el problema de construir una octava cromática a partir de una nota, la serie de 12 semitonos que forman una octava. Esto no es tan sencillo. Los intervalos entre semitonos en la tabla de números de tono (Apéndice E) no son uniformes; tampoco son uniformes los intervalos de frecuencia. Así pues, tenemos que calcular la frecuencia correcta de cada nota y convertirla en el número de tono asociado.

Los cálculos se realizan mediante una ecuación. El apéndice 7 del manual del usuario da una fórmula, *pero es incorrecta*. La fórmula correcta es la siguiente:

$$\text{frecuencia} = 440 * (2^{(\text{octava} + ((n - 10) / 12)))}$$

donde «octava» es el número de octava de las ocho disponibles (0-7, véase el apéndice E) y «n» es el número de orden de la nota (DO=1, DO#=2, ...).

Una vez calculada la frecuencia, el número de tono se obtiene mediante la fórmula

$$T = \text{ROUND}(125000 / \text{frecuencia})$$

El programa que necesitamos es, por lo tanto, el 5.1.

### Programa 5.1

```

10 FOR num=1 TO 12
20 READ a(y)
30 frec=440*(2^(0+((a(y)-10)/12)))
40 tono=ROUND(125000/frec)
50 SOUND 1,tono,35,15
60 NEXT
70 DATA 1,2,3,4,5,6,7,8,9,10,11,12

```

La línea 70 contiene los 12 números que representan los semitonos de una octava, empezando por 1 (DO). La línea 30 convierte esos números en frecuencias, y la línea 40 convierte las frecuencias en números de tono.

### ESCALAS

Partiendo de esta base, no es difícil crear una escala, digamos la DO mayor. En el capítulo 4 hemos explicado la regla para generar escalas mayores; la sucesión de notas es tono, tono, semitono, tono, tono, tono, semitono.

Todo lo que tenemos que hacer es modificar la línea de datos:

```
70 DATA 1.3.5.6.8.10.12.13
```

y cambiar la línea 10:

```
10 FOR num=1 TO 8
```

Ejecute el programa y oirá la escala de DO mayor, desde DO media hasta la DO de la octava siguiente. Los datos de la línea 70 se comprenden fácilmente. El número 1 representa DO; los restantes, los sucesivos tonos o semitonos de la escala. Para pasar de una nota al semitono siguiente hay que incrementar el número en una unidad; para pasar al tono siguiente, en dos unidades.

### MELODÍAS

Ahora ya estamos en condiciones de convertir melodías conocidas en programas para el Amstrad. Teclee y ejecute el programa 5.2. La música debería serle conocida; si quiere averiguar de qué melodía se trata tendrá que teclear el programa, porque yo no voy a decírselo.

**Programa 5.2**

```

10 tempo=2.5
20 RESTORE 90
30 FOR x=1 TO 37
40 READ tono,duracion
50 freq=440*(2^(0+((tono-10)/12)))
60 numtono=ROUND(125000/freq)
70 SOUND 1,numtono,duracion*tempo,15
80 NEXT
90 DATA 27,10,29,10,25,10,22,20,24,10,20,20
100 DATA 15,10,17,10,13,10,10,20,12,10,8,20
110 DATA 3,10,5,10,1,10,-2,20,0,10,-2,10,-3,10,-
4,40
120 DATA 3,10,4,10,5,10,13,20,5,10,13,20,5,10,13
,40
130 DATA 13,10,15,10,17,10,13,10,15,10,17,20,12,
10,15,20,13,40

```

¿La reconoce? Quizá se esté preguntando cómo funciona este programa. El tempo se puede modificar jugando con la línea 10; cuanto menor sea el número, más rápida será la ejecución, y viceversa. La línea 70 contiene la instrucción que genera el sonido. Observe que el tercer parámetro de SOUND es una combinación de «duración» y «tempo». La duración se lee de las líneas de datos, en las que se han incluido dos datos para cada nota: el primero es el número de tono y el segundo el de duración. El significado de los números de duración no nos preocupa por el momento; lo estudiaremos en el capítulo 10. Ahora sólo vamos a ocuparnos de los números de tono.

El núcleo del programa, líneas 50 a 70, es igual al del programa 5.1. La melodía ha sido programada poniendo en las líneas de datos los números de las notas y convirtiéndolos en números de tono. Los números de las notas son los que figuran en el apéndice E, suponiendo que se cuentan las notas a partir de DO media y que a esa octava se le dé el valor 0 (véase la línea 50 del programa). En cuanto adquiera cierta soltura con este método, verá qué fácil le resulta convertir melodías en programas para el Amstrad.

Para practicar, elija una melodía sencilla, que conste de notas simples (es decir, sin acordes), y convierta las notas en números para las líneas de DATA. No tenga en cuenta, por ahora, la duración de las notas, para la cual puede poner un valor típico en la línea 70. Al no haber programado la duración, la melodía sonará un poco rara, pero habrá conseguido el objetivo que pretendemos, que es practicar la conversión de partituras a programas.

Ahora vamos a abandonar la música momentáneamente, pues vamos a estudiar las dos instrucciones de sonido restantes: ENV y ENT.



---

## La envolvente de volumen

La instrucción de envolvente de volumen (ENV) permite al usuario controlar la variación del volumen con el tiempo para un sonido dado. Se la utiliza para modificar el efecto de la instrucción SOUND. Al igual que ésta, tiene una serie de parámetros que la hacen parecer complicada a primera vista. Pero, dado que muchos de ellos son análogos entre sí, no es tanto lo que hay que explicar.

La instrucción tiene la forma siguiente:

ENV n,P1,Q1,R1,P2,Q2,R2,P3,Q3,R3,P4,Q4,R4,P5,Q5,R5

¡No se asuste! En realidad, los únicos parámetros que tenemos que entender son los cuatro primeros. Pero veamos antes qué forma tiene el sonido.

### LA FORMA DEL SONIDO

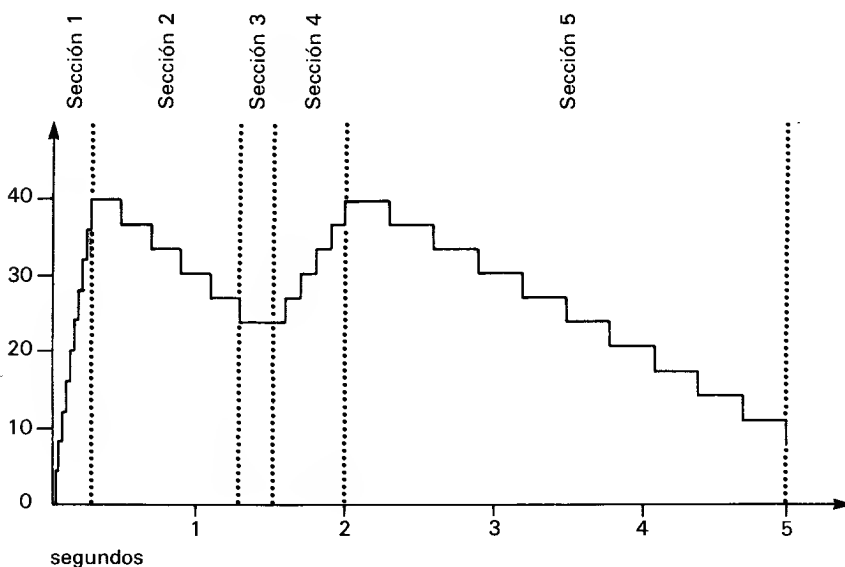
Si no nos organizamos, la instrucción ENV puede hacernos perder mucho tiempo. Puesto que podemos jugar con 15 parámetros, las combinaciones posibles son tantas que no podemos hacer pruebas al azar. Vale la pena, por lo tanto, dedicar unos minutos a analizar el sonido que queremos crear.

Todos los sonidos pueden ser caracterizados por la gráfica de su amplitud en función del tiempo (véase la figura 6.1). Cada sonido tiene una forma característica, que es la que tenemos que imitar al escribir la instrucción ENV. Teclee el programa 6.1 y Pruébalo.

#### Programa 6.1

```
10 ENV 1,10,4,3,5,-3,20,1,0,20,5,3,10,10,-3,30
20 SOUND 1,478,0,0,1,0,0
```

El sonido que acaba de escuchar está controlado por la instrucción ENV. La línea 20 define el tono, pero la duración especificada es 0, y esto indica a la máquina que debe consultar la instrucción ENV para averiguar la duración y otras características del sonido. También se ha especificado 0 como número de volumen, pues éste está descrito con más detalle en ENV. El parámetro EV es 1, para indicar que la envolvente de volumen que se debe utilizar es la número 1, que es la que hemos definido en la línea 10. Analicemos la forma de la figura 6.1.



**Fig. 6.1** Ejemplo de envolvente de volumen.

La forma está dividida en cinco secciones, cada una de las cuales se caracteriza por tres parámetros. En la figura 6.2 se muestra la relación entre las cinco secciones y los correspondientes parámetros de ENV.

Los tres parámetros de cada sección son los siguientes:  $P_n$  (donde  $n$  es el número de la sección), que es el número de escalones;  $Q_n$ , que es la altura de cada escalón; y  $R_n$ , que es la duración de cada escalón. En toda instrucción ENV hay que especificar al menos una sección.

El número de escalones es un entero comprendido entre 0 y 127. Cada uno de los escalones es de duración fija, caracterizado por  $R_n$ , que se expresa en unidades de centésimas de segundo; el margen de  $R_n$  es de 0 a 255. Para el parámetro  $Q_n$ , altura de los escalones, el margen es de -128 a +127; un número negativo indicará que la amplitud es decreciente.

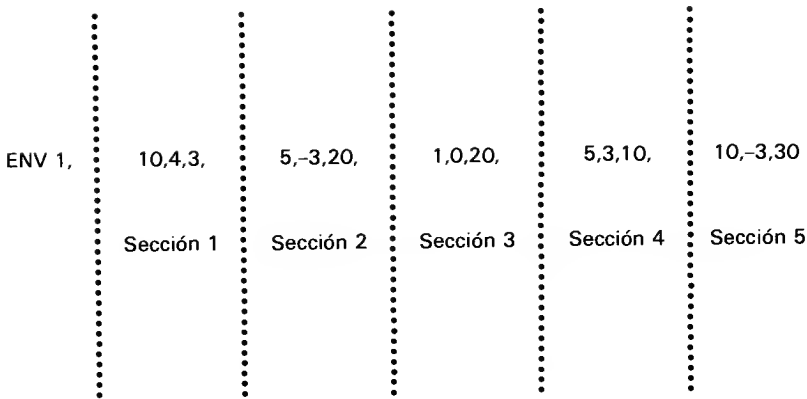


Fig. 6.2 Las cinco secciones de la envolvente de volumen.

La envolvente de la figura 6.1 dura 5 segundos. En efecto, si multiplicamos el número de escalones por la duración de cada uno de ellos y sumamos los resultados para las cinco secciones obtenemos:

$$(3*10)+(20*5)+(20*1)+(10*5)+(30*10)=500$$

Como las unidades son centésimas de segundo, 500 equivaldrá a 5 segundos.

Una vez definida una envolvente, el conjunto de parámetros queda almacenado en el generador de sonido, en espera de que más tarde utilicemos la envolvente. Si se quiere cancelar el efecto de una envolvente en las instrucciones SOUND, se la debe redefinir sin especificar la forma de las secciones, es decir, de la siguiente manera:

ENV 1

con lo que la envolvente queda desactivada.

## ENV EN LA PRÁCTICA

A pesar de que he dicho que es necesario estudiar la forma del sonido que se quiere generar antes de ponerse a escribir la instrucción ENV, voy a hacer una excepción. Para darle una idea de los efectos de los parámetros de ENV, el programa 6.2 le permite definirlos con facilidad; sólo hemos incluido una sección.

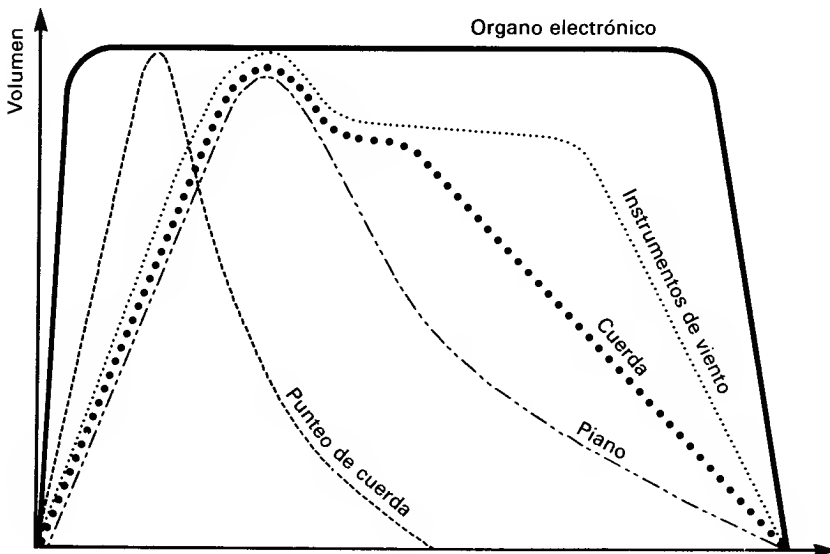
**Programa 6.2**

```

10 CLS
20 PRINT"Elija numero de saltos (0-127)"
30 INPUT numsaltos
40 PRINT"Elija tama\o del salto (-128 a +127)"
50 INPUT tamsalto
60 PRINT"Elija tiempo de pausa (0-255)"
70 INPUT tiempopausa
80 ENV 1,numsaltos,tamsalto,tiempopausa
90 SOUND 1,240,15,15,1,1,0
100 ENV 1
110 PRINT"Pulse una tecla para continuar"
120 x$=INKEY$:IF x$="" THEN 120
130 GOTO 10

```

Experimente un poco con estos números. Siempre es posible que descubra algún sonido interesante.



**Fig. 6.3**



## INSTRUMENTOS

Es probable que el lector pretenda imitar los sonidos de los instrumentos musicales. Dentro de ciertos límites, esto es posible; pero no olvide que estamos trabajando con unos circuitos electrónicos que tienen limitaciones inevitables. Siempre que alguien diseña una envolvente que cree que suena como un violín, hay otro que piensa que en realidad suena como un clarinete. Por esta razón no voy a dar una lista de envolventes predefinidas, sino solamente sugerencias para que usted las diseñe a su gusto.

En la figura 6.3 se indica la forma de onda de las notas generadas por diversos instrumentos musicales. Trate de imitar la forma de la curva al escribir la instrucción ENV y «afínela» hasta que suene a su gusto.

El ajuste más fino del sonido puede requerir que controle también la forma de variar la frecuencia de una nota a lo largo de su ejecución. Para ello sirve la instrucción ENT, que explicaremos en el capítulo siguiente.



---

## La envolvente de tono

Si, como espero, usted ha entendido cómo funciona la envolvente de volumen, este capítulo le resultará muy fácil. La envolvente de tono, ENT, se programa prácticamente igual que la de volumen, aunque, evidentemente, su acción se ejerce sobre el tono de la nota, no sobre la amplitud.

La instrucción ENT controla las variaciones de tono de las notas, las cuales crean un efecto pulsátil denominado *vibrato*. Estas pequeñas variaciones de tono afectan a los armónicos de la onda sonora y en muchos instrumentos musicales son las responsables de su singularidad.

### CONFORMACIÓN DE LOS TONOS

Los parámetros de ENT son fáciles de comprender: están dispuestos en grupos, igual que en la instrucción ENV. Son los siguientes:

ENT n,T1,V1,W1,T2,V2,W2,T3,V3,W3,T4,V4,W4,T5,V5,W5

donde «n» es el número de la envolvente. Al igual que en el capítulo anterior, he respetado las letras que se utilizan en el manual del Amstrad para designar los parámetros, a pesar de que no tienen relación alguna con las magnitudes que representan.

La envolvente de tono se puede dibujar en una gráfica del tono en función del tiempo. La instrucción ENT especifica la variación temporal del tono.

En el programa 7.1 definimos una envolvente típica utilizando cuatro de las cinco secciones posibles. Si ejecuta el programa tal como está, oirá el tono no controlado por ENT. Elimine la línea 15, la cual cancela la definición de la envolvente.

Programa 7.1

```
10 ENT 1,65,5,1,10,-2,10,10,2,5,30,-5,1
15 ENT 1
20 SOUND 1,478,50,15,0,1,0
```

Lo que acaba de escuchar es el efecto de la envolvente de tono sobre la nota. La figura 7.1 describe qué está ocurriendo, tanto en ENT como en SOUND.

Otro ejemplo de envolvente de tono es el que se muestra en la figura 7.2. En cada sección hay tres parámetros: Tn, que es el número de escalones;

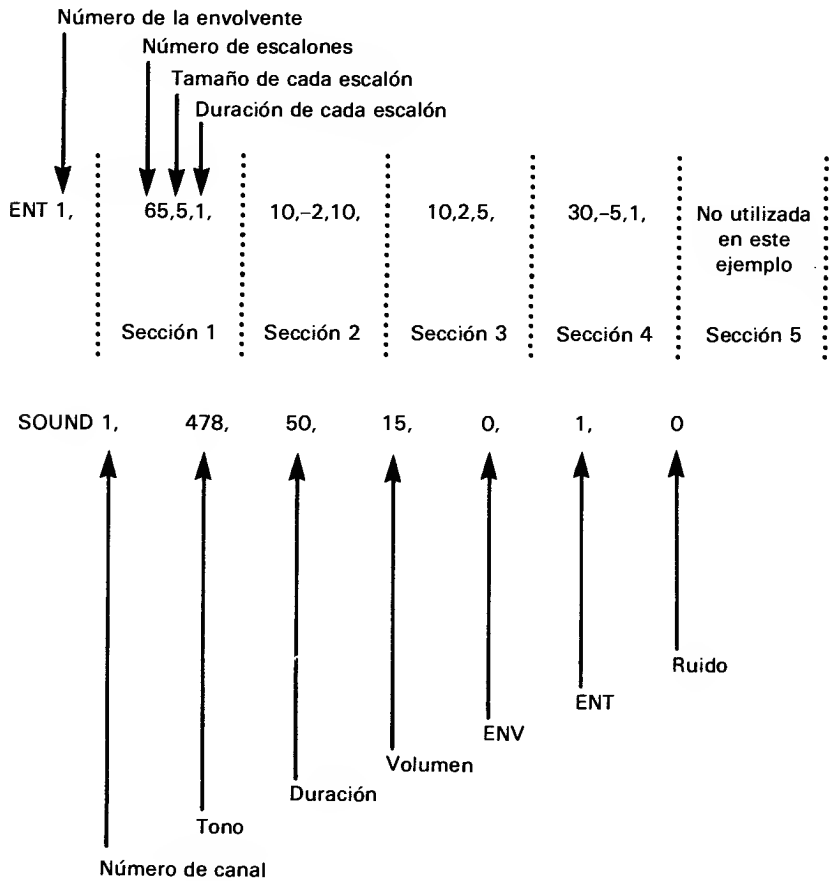


Fig. 7.1 Relación entre las instrucciones ENT y SOUND y secciones de una envolvente de tono.

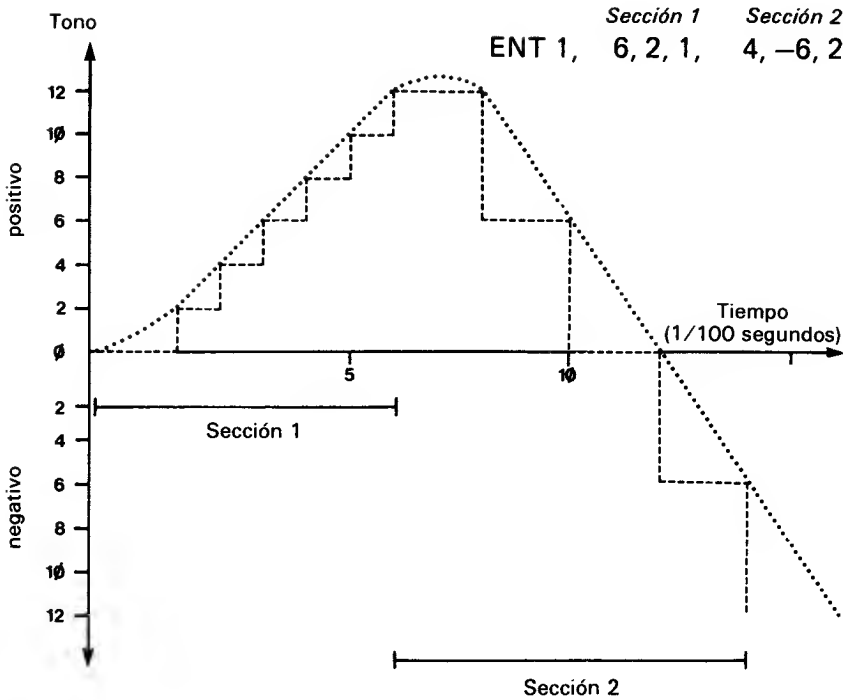


Fig. 7.2 Ejemplo de envolvente de tono.

$V_n$ , que es la altura de los escalones; y  $W_n$ , que es la duración de cada uno de ellos. Como ocurría con ENV, es necesario definir al menos una sección, salvo cuando se desee cancelar una envolvente. Si se empieza a definir una sección, hay que completarla, pues el ordenador no admite una instrucción en la que falte algún parámetro.

Los márgenes de valores de los parámetros ENT son los mismos que en ENV en cuanto a altura y duración de los escalones; en cambio, el número de escalones puede variar entre 0 y 239. Además, el primer parámetro,  $n$ , que identifica la envolvente, sirve para indicar si la envolvente ha de repetirse. En efecto, poniendo un número negativo, la envolvente se repite durante todo el tiempo en que la nota está sonando. Este sirve para simular el efecto *trémolo*.

Si desea experimentar con ENT, teclee el programa 7.2, que funciona de forma análoga a la del programa 6.2.

**Programa 7.2**

```

10 CLS
20 PRINT"Elija numero de escalones (0-239)"
30 INPUT numesc
40 PRINT"Elija tama\o de escalones (-128 a +127)"
50 INPUT tamaesc
60 PRINT"Elija duracion de escalones (0-255)"
70 INPUT duraesc
80 ENT 1,numesc,tamaesc,duraesc
90 SOUND 1,240,100,15,0,1,0
100 ENT 1
110 PRINT"Pulse una tecla para continuar"
120 x$=INKEY$:IF x$="" THEN 120
130 GOTO 10

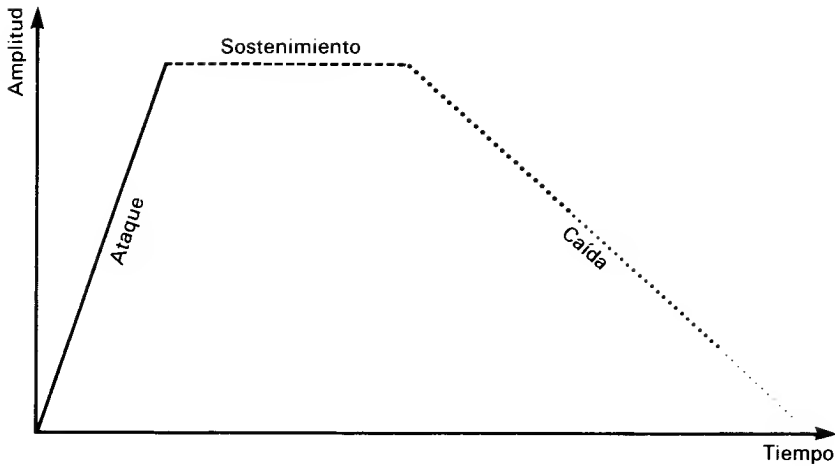
```

**ATAQUE, SOSTENIMIENTO Y CAÍDA**

Y con esto casi hemos terminado nuestro estudio de las envolventes. La inclusión de las envolventes en BASIC facilita el control más completo del generador de sonido: al definir una envolvente se puede especificar la forma de ataque, sostenimiento y caída de las notas. Si no conoce esta terminología, no se alarme; de hecho, hemos estado manejando estos conceptos en los dos últimos capítulos. Cuando diseñamos envolventes, estamos aumentando, reduciendo o manteniendo constantes la amplitud o el tono. Cuando oímos un sonido cualquiera, observamos que el volumen aumenta (generalmente de prisa) al principio: es la fase de ataque. Una vez alcanzada la amplitud máxima, esta se mantiene constante durante algún tiempo (fase de sostenimiento) y luego decrece hasta extinguirse (fase de caída). Todos los sonidos constan de estas tres fases, pero con duraciones y formas diferentes, que los caracterizan. Estudiemos estos factores más detenidamente, haciendo referencia a la envolvente de volumen.

La fase de ataque es la primera. Determina cuánto tiempo tarda el sonido en alcanzar su nivel máximo. El piano, por ejemplo, tiene una fase de ataque muy corta, ya que su sonido se produce al golpear las cuerdas con un mazo. Piense en el sonido de otros instrumentos y trate de imaginar cómo será su fase de ataque.

La fase de sostenimiento mantiene más o menos constante el volumen del sonido. En la fase de caída el volumen se va reduciendo paulatinamente hasta cero. En la figura 7.3 se ilustran estas tres fases.



**Fig. 7.3** Las tres fases de un sonido.

El generador de sonido del Amstrad no puede producir buenas imitaciones de los instrumentos musicales, ya que el tono que genera no es una onda sinusoidal pura, sino una onda cuadrada con sus armónicos inherentes. Pero no se desanime. Se pueden conseguir aproximaciones aceptables mediante el juicioso control de las envolventes de tono y de volumen. Tras cierto trabajo experimental, el lector podrá ir formando su sonoteca de instrumentos musicales.

Así quedan explicadas las principales instrucciones que intervienen en la generación de sonido en el ordenador Amstrad. En los capítulos que siguen continuaremos explorando el generador de sonidos en busca de todo tipo de efectos, desde música hasta invasores extraterrestres.





---

## Efectos especiales

El término *sonidos* que hemos puesto en el título de este libro engloba tanto la música como los ruidos y todos los efectos sonoros. Y, efectivamente, el Amstrad puede generar, además de música, efectos sonoros espectaculares que el lector puede incorporar a sus programas.

El Amstrad puede ser ruidoso como el que más; podemos utilizar sus sonidos en la forma que nos convenga. La primera aplicación que se nos viene a la mente es el diseño de programas de juegos, en los que el sonido puede ser fundamental.

Los juegos de ordenador no serían tan atractivos si no fuera por los ruidos de choques, explosiones y disparos. Dé rienda suelta a su imaginación y pronto estará creando efectos sonoros dignos de la capacidad gráfica de este ordenador.

### AVENTURA SONADA

*Un viaje por los canales sonoros del tiempo y el espacio*

Estamos deslizándonos suavemente por el espacio a hipervelocidad 7.6 cuando nuestro monitor intergaláctico ultrasensible detecta que se nos acerca una nave enemiga a velocidad tetradimensional. Pulsamos el botón de alerta general y...

... teclee el Programa 8.1.

#### Programa 8.1

```
10 ENT 1,80,-4,1
20 SOUND 1,478,50,15,0,1,0
30 GOTO 20
```

La alerta ha sonado; los tripulantes del Federation Star Amcruiser corren hacia sus puestos de combate, donde quedan a la espera del primer signo de acción hostil. De pronto la nave enemiga desaparece de nuestras pantallas y todo queda tranquilo. Llegamos a dudar de si verdaderamente existe el aparato enemigo, cuando de repente la impresora ultramatricial vuelve a la vida y se pone a escribir...

### Programa 8.2

```
10 FOR tipo=1 TO 45
20 SOUND 1,901,7,15,0,0,1
30 FOR retardo=1 TO 100:NEXT
40 NEXT
```

El torrente de caracteres escritos por la impresora es incomprensible para los humanos. ¿Por qué no utilizan comunicación electromagnética? Como si nos hubieran leído el pensamiento, el comunicador que tenemos en la mano despierta de su largo sueño...

### Programa 8.3

```
10 FOR repetir=1 TO 6
20 radio=INT(RND*20)
30 SOUND 1,radio,100,15,0,0,1
40 NEXT
```

Pero nuestro comunicador no puede sintonizar la frecuencia de los enemigos. ¿Nos quedaremos sin saber qué está ocurriendo? Probablemente, pues este sueño está tocando a su fin. Nuestra única esperanza es recurrir al ordenador de a bordo. Tecleamos rápidamente nuestro problema y el ordenador lo procesa.

### Programa 8.4

```
10 FOR repetir=0 TO 140
20 calcula=INT(RND*150)
30 SOUND 1,calcula,3,15,0,0,0
40 NEXT
```

Esperamos impacientes la respuesta. Pasan unos segundos interminables; nos devoramos las uñas... ya está la respuesta en la pantalla: «FIN DE LA PARTIDA. INTRODUCIR UNA MONEDA DE 25 PESETAS PARA SEGUIR JUGANDO.»

Bien, como escritor de ciencia ficción no seré gran cosa, pero he demostrado que en toda narración se pueden encajar sonidos. Le sugerimos que reúna los programas anteriores en uno solo para hacer sonar el efecto adecuado en cada momento del cuento.

Añadiendo ruido al sonido, mediante el último parámetro de SOUND, podemos convertir el programa 8.4 en el momento final de la caída de una bomba. Teclee el programa 8.5.

### Programa 8.5

```

10 FOR cae=50 TO 150
20 SOUND 1,cae,3,15,0,0,0
30 NEXT
40 FOR repite=0 TO 45
50 calcula=1
60 SOUND 1,calcula,3,15,0,0,31
70 NEXT

```

Todo lo que se requiere es un poco de imaginación y ganas de experimentar. Cuando se buscan efectos sonoros es conveniente experimentar con una instrucción SOUND y las instrucciones de definición de envolventes. El siguiente programa es un generador de envolventes de volumen. Tecléelo y ejecútelo.

### Programa 8.6

```

10 :
20 REM Inicializar programa
30 :
40 ZONE 40
50 ENV 1
60 CLS
70 :
80 REM Definir menu
90 :
100 PRINT"GENERADOR DE ENVOLVENTES DE VOLUMEN"
110 PRINT,, "1. Envolvente de volumen"
120 PRINT,, "Elija"
130 a$=INKEY$
140 a=VAL(a$):IF a<>1 THEN 130
150 PRINT a

```

```

160 ON a GOTO 170
170 CLS
180 :
190 REM Elegir numero de secciones
200 REM de la envolvente especificada
210 :
220 PRINT"ENVOLVENTE NUMERO 1"
230 PRINT,,,"Introduzca numero de secciones (max
. 5)"
240 INPUT secciones
250 FOR sec=1 TO secciones
260 PRINT"Introduzca numero de escalones (0-127)
"
270 INPUT ne(sec)
280 PRINT,,,"Introduzca altura de los escalones (
-128 a +127)"
290 INPUT te(sec)
300 PRINT,,,"Introduzca duracion de los escalones
(0-255)"
310 INPUT de
320 :
330 REM Escribir parametros en la instruccion
340 REM de envolvente de volumen
350 :
360 ENV 1,ne(1),te(1),de(1),ne(2),te(2),de(2),ne
(3),te(3),de(3),ne(4),te(4),de(4),ne(5),te(5),de
(5)
370 :
380 REM Ir a la siguiente seccion, si la hay
390 :
400 NEXT
410 :
420 REM Captar parametros para
430 REM instruccion SOUND
440 :
450 CLS
460 PRINT"P A R A M E T R O S   D E   S O N I D
O"
470 PRINT"Que canal?"
480 INPUT canal
490 PRINT"Duracion?"
500 INPUT duracion
510 PRINT"Ruido (0 a 31)?"
520 INPUT ruido
530 :

```

```

540 REM Ejecutar SOUND
550 :
560 SOUND canal,478,duracion,1,1,0,ruido
570 :
580 REM Escribir parametros
590 REM de envolvente
600 :
610 CLS
620 PRINT,,"Los parametros de la envolvente son
los siguientes:"
630 PRINT"Envolvente 1"
640 FOR x=1 TO 5
650 PRINT"NS";x;"=";ne(x)
660 PRINT"TS";x;"=";te(x)
670 PRINT"TP";x;"=";de(x)
680 NEXT
690 PRINT"Quiere oirla otra vez?"
700 a$=INKEY$:IF a$="" THEN 700
710 IF a$="S" OR a$="s" THEN 560
720 PRINT"Quiere cambiar los parametros?"
730 PRINT"del sonido"
740 a$=INKEY$:IF a$="" THEN 740
750 IF a$="S" OR a$="s" THEN 450
760 GOTO 40

```

El programa empieza por ofrecer un menú. En esta versión del programa la única opción es la generación de envolventes de volumen. Cuando haya utilizado el programa y comprendido su funcionamiento, puede añadir la otra opción, la de la envolvente de tono, o cualquier otra mejora que se le ocurra para facilitar la programación de efectos sonoros. Una vez dentro del núcleo del programa, éste le pregunta cuántas secciones quiere definir para la instrucción ENV. Si en ese momento pulsa ENTER, no se define ninguna envolvente. Puede elegir cualquier número entre 1 y 5.

A continuación el programa pide los tres parámetros de cada sección, indicando los márgenes permitidos. Finalmente, el programa ofrece las opciones de los parámetros de la instrucción SOUND y pregunta si se quiere añadir ruido. Se ejecuta el sonido así definido y se visualizan los parámetros de la envolvente. Otras opciones permiten repetir el sonido o modificar los parámetros de SOUND.

El programa debería ser fácil de comprender, pues las instrucciones más importantes están explicadas con líneas REM. Este programa le ayudará a ahorrar tiempo al permitirle probar, con un mínimo esfuerzo, distintos parámetros hasta que dé con el sonido que está buscando.

Para darle un punto de partida, en la tabla 8.1 le facilito los parámetros

de algunos de los sonidos que yo he investigado. Tenga en cuenta que algunos de estos sonidos pueden requerir alguna manipulación adicional que el programa 8.6 no permite.

**Tabla 8.1** Parámetros de envolventes de volumen (sólo tres secciones) y de la instrucción SOUND.

Sonido	Sección 1			Sección 2			Sección 3			Selecc. de canales	Duración	Ruido
	NE1	AE1	DE1	NE2	AE2	DE2	NE3	AE3	DE3			
Disparos	12	13	14	15	16	17	18	19	20	7	145	30
Instrumento de cuerda	12	13	14	15	16	17	18	19	20	7	145	0
Golpes en puerta o tambor	23	-68	3	—	—	—	—	—	—	7	120	12
Explosión	15	5	1	15	-1	30	—	—	—	7	120	20

## PERCUSIÓN

Finalmente, antes de abandonar el tema de los efectos sonoros, debemos hablar de los sonidos de percusión, que indudablemente interesarán a quienes deseen programar música. El canal de ruido, en conjunción con las instrucciones ENV y ENT, puede dar diversos efectos de pseudopercusión, desde el redoblante hasta el bombo. El programa 8.7 es un sencillo ejemplo de estas posibilidades.

### Programa 8.7

```

10 ENV 1,23,-68,3
20 FOR duración=1 TO 50
30 SOUND 7,748,duración,15,1,1,1
40 NEXT

```

Y con esto terminamos nuestra breve incursión por el planeta de los efectos sonoros. Usted puede pasar muchas horas probando y seleccionando sonidos. Con toda seguridad, descubrirá alguno que pueda incluir en sus programas, ya sea para hacerlos más atractivos, por mera diversión o para irritar a los vecinos.

## Música, maestro

Mucho hemos avanzado desde el principio del libro. Ha llegado la hora de convertir al lector en un pianista. Uno de los objetivos de esta obra es ayudarle a crear música, pero hasta ahora todo lo que le he dejado hacer es poner los datos de las notas en líneas DATA. Este es un procedimiento lento y difícil cuando se quieren ejecutar piezas relativamente largas.

Lo que necesitamos es una forma de convertir el teclado del Amstrad en un teclado musical. No es un teclado ideal, pero al menos es un conjunto ordenado de teclas a las que podemos hacer que correspondan determinadas notas. ¿Cómo podemos convertir un teclado de ordenador en un instrumento musical? Si piensa que para ello hace falta un gran programa, está equivocado. Con ocho líneas tenemos suficiente.

### UN SINTETIZADOR

#### *Teclado de piano en el Amstrad*

Observe las teclas de su Amstrad. No tiene mucho en común con las de un piano, ¿verdad? Sólo se parecen en que pueden ser pulsadas independientemente unas de otras. Nos basaremos en este hecho. En el piano las teclas están dispuestas en una larga fila que cubre varias octavas, de tono menor a mayor. El teclado del ordenador, en cambio, consta de varias filas; vamos a intentar asignar a una de ellas una octava y después veremos dónde poner las restantes. Tomemos la primera fila de teclas literales (QWERTY...); éstas serán nuestras teclas «blancas». Las negras serán las de la fila superior, las teclas numéricas. Vamos con el programa. Vaya tecleando las líneas a medida que se las vaya explicando, pero no ejecute el programa mientras no esté completo. La primera línea es la 30:

```
30 w$=INKEY$:IF w$="" THEN 30
```

Como puede observar, esta instrucción se repite hasta que se pulsa una tecla cualquiera, momento en el que el programa pasa a la línea siguiente. Ahora tenemos que decirle a la máquina qué teclas vamos a utilizar. Esto es lo que hacemos en la línea 10, en la que la cadena s\$ contiene las teclas correspondientes a todas las notas de la octava, incluidos los semitonos. Como puede ver en la definición de s\$, las letras están en minúsculas; así pues, cuando ejecute este programa, cerciúrese de que no están bloqueadas las mayúsculas en CAPS LOCK.

```
10 s$="q2w3er5t6y7ui9oOp"
```

La instrucción que produce el sonido está en la línea 70:

```
70 SOUND 1,tono,15,15
```

La variable «tono» todavía no está definida. Para obtener los valores correctos necesitamos las fórmulas del capítulo 5:

```
50 frecuencia=440*(2 (1+((nota-10)/12)))
60 tono=ROUND(125000/frecuencia)
```

Recuerde que el valor de la variable «nota» de la línea 50 determina qué nota se debe ejecutar (DO=1, RE=3, ...). Puesto que hemos decidido que la primera sea la tecla «q», ésta dará la nota DO. ¿Por qué? Veamos la línea 40:

```
40 nota=INSTR(s$,w$)
```

Aquí la función INSTR determina si la tecla pulsada, w\$, coincide con alguno de los caracteres de s\$. Si es así, asigna a «nota» el valor numérico de la nota. Por ejemplo, si la tecla pulsada es «q», «nota» toma el valor 1, que corresponde a la nota DO.

Lo que necesitamos ahora es incluir este programa en un bucle que se repita indefinidamente. Para ello recurrimos a la instrucción WHILE ... WEND, líneas 20 y 80:

```
20 WHILE x=0
80 WEND
```

El programa ya está completo. Cuando lo ejecute tendrá a su disposición un teclado musical que empieza por la nota DO (letra «q») y se extiende hacia la derecha cubriendo más de una octava. Si quiere limitarlo a una octava, suprima de la definición de s\$ los cuatro últimos caracteres.



En la figura 9.1 se muestra la correspondencia entre el teclado del Amstrad y el del piano. Recuerde que en el apéndice E se dan las tablas completas de frecuencia, números de tono y números de las notas. Cambiar la octava es muy fácil. En la ecuación de la línea hemos puesto 1 como número de octava. Si ponemos 0, el efecto es bajar el tono de todas las notas a la octava inmediatamente inferior.

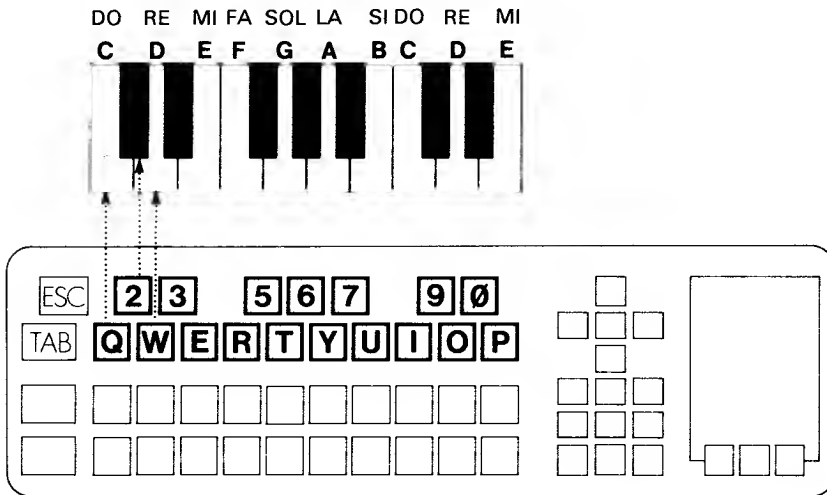


Fig. 9.1 Relación entre el teclado del Amstrad y el del piano.

Y ahora, manos al teclado. Por si usted es realmente lego en música, en la figura 9.2 he incluido una serie de notas que puede ejecutar; para mayor facilidad, he puesto las letras del teclado, no los nombres de las notas. No doy ninguna indicación sobre el ritmo, pero eso lo aportará usted en cuanto reconozca la melodía.

## MÁS OCTAVAS

### *Ampliación del teclado*

Como dijimos antes, el margen cubierto por el teclado se puede ampliar a otras octavas. Esto se puede hacer de varias formas; la que yo prefiero consiste en utilizar las dos últimas filas del teclado para que sean continuación de las dos primeras.

Q	W	E	Q	E	Q	E	W	E	R
R	E	W	R	E	R	T	E	T	E
T	R	T	Y	Y	T	R	Y	T	Q
W	E	R	T	Y	Y	W	E	R	T
Y	U	U	E	R	T	Y	U	I	I
U	Y	Y	R	U	T	I	Q	W	E
R	T	Y	U	I	Q				

Fig. 9.2 Melodía misteriosa.

Para realizar la ampliación, añade las siguientes líneas al programa:

```

15 octava$="zxsxdcvgbhnjm,l.:/"
65 IF nota=0 THEN GOSUB 90
90 nota=INSTR(octava$,w$)
100 frecuencia=440*(2^(0+((nota-10)/12)))
110 tono=ROUND(125000/frecuencia)
120 RETURN

```

Su teclado cubre ahora dos octavas. Así, para ejecutar la escala de DO mayor en dos octavas pulse las siguientes teclas: zxcvbnm,wertyui.

Los dos grupos de letras se extienden hasta un poco más allá del límite de la octava correspondiente, de modo que algunos semitonos se pueden ejecutar con dos teclas distintas.

El sintetizador que hemos construido es un buen instrumento para jugar y experimentar. Este programa es una introducción al uso de sintetizadores; espero que le sirva como base para ampliarlo e incluir otros efectos sonoros. Para ello se requiere manejar las envolventes, además de un poco de imaginación. ¡Buena suerte!

---

## Ritmo

Recuerde que en el capítulo 5 explicamos cómo convertir música en números para ponerlos en líneas DATA. Hablamos de la conversión de los tonos, pero dejamos pendiente la de la duración de las notas. En este capítulo vamos a estudiar cómo convertir esa notación musical en números que el Amstrad pueda entender.

En música, el tiempo es un factor obviamente muy importante. De él depende el ritmo y el tempo.










Pero dejemos las cosas claras. Recuerde que, en la notación musical, las corcheas, fusas y similares caracterizan la duración de las notas; pero no en términos absolutos, sino en forma de diferencias entre unas y otras. No dan, pues, información sobre el tempo, que es el responsable de la velocidad global a la que se interpreta la pieza musical. No confunda estos dos conceptos, que son independientes entre sí.

Volviendo al capítulo 5, recordará que cuando interpretó la melodía secreta del programa 5.2, no sólo era correcta la velocidad global, sino que cada nota tenía la duración adecuada. Cuando traduje la música a números, tuve que utilizar una tabla de conversión de notas (corcheas, semicorcheas, etc.) a tiempo. Al programar música es necesario asegurarse de que la duración de cada nota es correcta en relación con la de las restantes. Vea la tabla 10.1, en la que figuran las duraciones relativas de las diversas notas.

Como sabemos, si una corchea dura 20, una negra tiene que durar el doble: 40. En la tabla se indican los parámetros de duración de las notas. Ahora se nos plantea otro problema. Las melodías se pueden ejecutar con tempos diferentes. Sería un trabajo agotador tener que cambiar los números de duración en las líneas de datos cada vez que se quisiera modificar el tempo de una melodía.

Esta es la razón por la que en el programa 5.2 incluíamos una línea que definía el tempo. Su valor se multiplica por cada uno de los números de

**Tabla 10.1** Números de duración de las notas musicales en el Amstrad.

<i>Nota</i>									
<i>Duración</i>	10	15	20	30	40	60	80	120	160

duración antes de introducirlos en la instrucción SOUND. De esta forma, para variar el tempo no tenemos más que modificar una línea del programa.

Partiendo de esta base, usted puede empezar a pensar en los diferentes ritmos posibles que quiera dar a su música. Experimente con rock and roll, vales, pasodobles, ... El ritmo es un factor muy importante en música, hasta el punto de poder cambiar la naturaleza de una melodía.

En este momento usted dispone de toda la información necesaria para convertir música en programas para el Amstrad. Es misión suya experimentar y traducir a datos numéricos sus ideas musicales.

---

## Campanas y silbidos

El espectro de los sonidos del Amstrad es muy amplio. Prácticamente no está limitado más que por su imaginación y por el tiempo que usted pueda dedicar a buscar nuevos sonidos, ruidos y efectos. Toda la información básica necesaria para explorar los canales sonoros de su ordenador está en este libro. Con ella usted está en condiciones de crear efectos sonoros y musicales.

La mitad del atractivo de escribir un programa que utilice sonido está en descubrir uno mismo los pitidos y ruidos necesarios. Si todavía está algo confuso por la multitud de parámetros y no entiende a la perfección el funcionamiento de alguna instrucción, tómesele con calma. Este libro está pensado para quienes gustan de experimentar. Juegue con las instrucciones todo lo que desee. Ésa es una forma de descubrir cómo funciona una instrucción.

También le he presentado las nociones de música en el Amstrad. Programas como el del sintetizador le darán una base firme sobre la que construir otras obras musicales más atractivas y complicadas. El tema de la música no ha quedado agotado, ni mucho menos. Espero que usted se sienta inspirado y con suficiente entusiasmo para continuar por su cuenta imitando instrumentos musicales, creando armonía para tres voces, simulando sonidos de tambores y muchas cosas más.

Al final del libro encontrará una serie de apéndices que pueden sacarle de algún apuro. No dude en consultarlos.

Tenga en cuenta que los sonidos que usted programa son una interpretación personal y que a otras personas pueden sonarle de forma diferente. Lo que está haciendo es aproximaciones a ciertos sonidos; algunas aproximaciones serán mejores que otras. Esto se deberá en ocasiones a las limitaciones del generador de sonidos. Su labor es forzar el generador hasta el límite de su capacidad; dónde se encuentra ese límite es algo que usted tendrá que descubrir.

No quiero terminar este libro sin ofrecerle unos efectos sonoros. ¿Qué mejor forma de terminar que con los ruidos a los que tengo que enfrentarme en mi jornada laboral? El desesperante teléfono que está comunicando...

```
10 SOUND 1,100,40,15
20 FOR pausa=0 TO 900:NEXT
30 GOTO 10
```

y, aunque no esté comunicando, no lo cogen...

```
10 ENV 1,100,122,1
20 SOUND 1,239,0,15,1,0,0
30 FOR x=0 TO 2000:NEXT
40 GOTO 20
```

Y, cómo no, el inevitable silbido de la locomotora:

```
10 FOR bocina=1 TO 3
20 SOUND 1,239,80,15,1,2,1
30 FOR pausa=0 TO 900:NEXT
40 NEXT
```

Bueno, el libro ha terminado. Pero para usted esto es el principio del viaje en el que va a desvelar los talentos ocultos de su Amstrad y la gran variedad de sonidos que puede producir.

## Apéndice A

---

### Palabras reservadas del BASIC de Amstrad

Este apéndice contiene un resumen de las instrucciones de BASIC del ordenador Amstrad. No pretende sustituir el manual del usuario ni dar una descripción detallada, sino solamente servir para refrescar la memoria o resolver alguna duda que pueda surgir en el momento de escribir un programa. Para más amplia información se debe consultar el manual del usuario.

ABS	Da el valor absoluto de su argumento.
AFTER	Se utiliza en combinación con GOTO para ejecutar una subrutina cuando haya transcurrido un tiempo especificado, controlado por los cronómetros internos.
AND	Operador lógico «y»
ASC	Convierte un carácter a código ASCII.
ATN	Arco tangente.
AUTO	Se utiliza para que el ordenador escriba automáticamente el número de línea cuando se están introduciendo programas.
BIN\$	Convierte un número decimal a forma binaria.
BORDER	Cambia el color del borde de la pantalla.
CALL	Invoca una subrutina en código de máquina.
CAT	Da la lista de los ficheros grabados en una cinta.
CHAIN	Carga y ejecuta un programa.
CHAIN MERGE	Mezcla con el programa residente en memoria un programa grabado en cinta y ejecuta el programa resultante.
CHR\$	Convierte un código ASCII en un carácter.
CINT	Redondea un número al entero más próximo.
CLEAR	Borra de la memoria todas las variables.
CLG	Borra la pantalla gráfica.
CLOSEIN	Cierra el fichero de cinta que está abierto para lectura.

CLOSEOUT	Cierra el fichero de cinta que está abierto para grabación.
CLS	Borra la pantalla.
CONT	Hace que continúe la ejecución de un programa que ha sido interrumpido.
COS	Coseno.
CREAL	Convierte el argumento en número real.
DATA	Señala las líneas de programa en que están contenidos los datos que van a ser leídos por la instrucción READ.
DEF FN	Define una función.
DEFINT	Define variables enteras.
DEFREAL	Define variables reales.
DEFSTR	Define variables literales.
DEG	Grados.
DELETE	Borra líneas de programa; por ejemplo, DELETE 10-60.
DI	Inhibe interrupciones.
DIM	Especifica las dimensiones de las matrices.
DRAW	Dibuja una recta que une los puntos especificados.
DRAWR	Dibuja una recta hasta una posición referida a la posición actual del cursor gráfico.
EDIT	Prepara una línea para su edición.
EI	Permite interrupciones.
ELSE	Utilizada en conjunción con IF ... THEN para dar un segundo curso de acción cuando la condición no se cumple; por ejemplo IF contador>20 THEN GOTO 50 ELSE GOTO 100.
END	Indica al ordenador dónde termina el programa.
ENT	Envolvente de tono.
ENV	Envolvente de volumen.
EOF	Función que indica si se ha alcanzado el final de un fichero.
ERASE	Borra las listas especificadas.
ERL	Función que da el número de la línea en la que se detectó el último error.
ERR	Función que da el número del último error detectado.
ERROR	Se utiliza en conjunción con ON para gestionar el tratamiento de los errores.
EVERY	Se utiliza en combinación con GOTO para ejecutar una subrutina a intervalos regulares, controlados por los cronómetros internos.
EXP	Calcula potencias del número <i>e</i> .
FIX	Elimina la parte decimal de los números.
FOR	Señala el principio de los bucles FOR ... NEXT.



FRE	Función que informa sobre el espacio libre de memoria.
GOSUB	Ordena la ejecución de la subrutina que empieza en la línea especificada.
GOTO	Envía el programa a la línea especificada.
HEX\$	Convierte un número decimal a forma hexadecimal.
HIMEN	Función que da posición de memoria más alta utilizable por BASIC.
IF	Principio de la instrucción IF ... THEN.
INK	Cambia la tinta especificada a un determinado color.
INKEY	Examina el teclado para detectar si una tecla determinada está siendo pulsada, ya sea sola o en combinación con SHIFT o CTRL.
INKEY\$	Si hay una tecla pulsada, da el carácter correspondiente.
INP	Función que da el valor recibido en la puerta especificada.
INPUT	Capta datos por el teclado.
INSTR	Determina si una cadena literal está contenida en otra.
INT	Convierte un número al entero más próximo.
JOY	Examina la situación de los joysticks.
KEY	Asigna una función a un carácter definible.
KEY DEF	Asigna un carácter a una tecla.
LEFT\$	Copia un tramo de una cadena literal empezando por su izquierda.
LEN	Determina la longitud de una cadena literal.
LINE INPUT	Análoga a INPUT, pero admite la presencia de signos de puntuación en las cadenas introducidas por el teclado.
LIST	Escribe en la pantalla las instrucciones de que consta el programa.
LOAD	Carga un programa de la cinta a la memoria del ordenador.
LOCATE	Lleva el cursor de texto a la posición especificada.
LOG	Logaritmo.
LOWERS\$	Convierte mayúsculas en minúsculas.
MAX	Halla el valor máximo.
MEMORY	Fija la más alta posición de memoria utilizable por BASIC.
MERGE	Mezcla un programa grabado en cinta con el residente en la memoria.
MID\$	Copia un tramo de una cadena literal tomado a partir de la posición especificada.
MIN	Halla el valor mínimo.
MODE	Selecciona el modo de pantalla (20, 40 u 80 columnas).

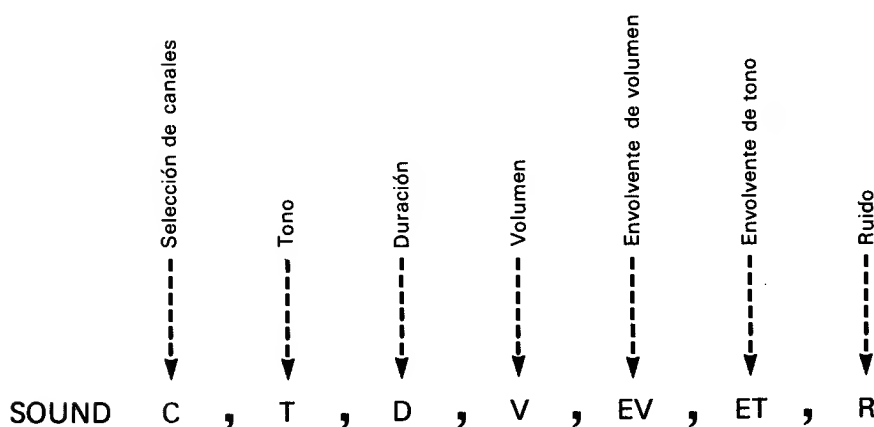
MOVE	Lleva el cursor gráfico a la posición especificada.
NEW	Borra la memoria del ordenador accesible a BASIC.
NEXT	Indica el final de un bucle FOR ... NEXT.
ON	Reconduce la ejecución del programa en función de un acontecimiento o del valor de una variable.
ON SQ GOSUB	Permite interrupciones cuando hay espacio libre en la cola de sonidos.
OR	Operador lógico «o».
OPENIN	Abre un fichero de cinta para su lectura.
OPENOUT	Abre un fichero de cinta para grabación.
ORIGIN	Fija el origen para el cursor gráfico.
PAPER	Asigna un tintero al papel (fondo)
PEEK	Lee el contenido de una posición de memoria.
PEN	Asigna un tintero a la pluma (color de los caracteres).
PI	Da el valor $\pi$ (3.14159).
PLOT	Ilumina el pixel especificado.
POKE	Introduce un valor en una posición de memoria.
PRINT	Escribe en la pantalla.
RAD	Radianes.
RANDOMIZE	Inicializa la sucesión de números aleatorios.
READ	Lee la información contenida en una línea DATA.
RELEASE	Libera los sonidos retenidos en una cola.
REMAIN	Inhibe un cronómetro interno.
RENUM	Renumerar las líneas de un programa.
RESTORE	Hace que el puntero de datos señale la línea especificada.
RESUME	Reanuda la ejecución del programa después de ON ERROR.
RETURN	Señala el final de una subrutina.
RIGHT\$	Copia un tramo de una cadena literal tomando caracteres de su derecha.
RND	Genera un número aleatorio.
ROUND	Redondea el argumento al número de decimales especificado.
RUN	Ejecuta el programa residente en la memoria.
SAVE	Graba el programa en cinta.
SGN	Da el signo del argumento.
SOUND	Véase todo este libro.
SPACE\$	Genera una cadena de espacios.
SPEED INK	Especifica la velocidad de parpadeo de las tintas.
SPEED KEY	Especifica la velocidad de repetición de las teclas.
SPEED WRITE	Especifica la velocidad de transferencia de datos a la cinta.

SPC	Escribe espacios en la pantalla.
SQ	Examina la situación de un canal de sonido.
SQR	Raíz cuadrada.
STEP	Especifica el salto en los bucles FOR ... NEXT.
STOP	Interrumpe la ejecución del programa.
STR\$	Convierte un número a forma literal.
SYMBOL	Redefine la forma de un carácter.
TAB	Controla la posición de escritura en la instrucción PRINT.
TAG	Permite escribir texto en la posición del cursor gráfico.
TAGOFF	Cancela la instrucción TAG.
TAN	Tangente.
TESTR	Da el número de tinta que hay en la posición del cursor gráfico especificada.
THEN	Utilizada en conjunción con IF.
TIME	Da el tiempo transcurrido desde que se encendió la máquina.
TO	Utilizada en la definición de un bucle FOR ... NEXT para especificar el margen de valores de la variable del contador.
TRON	Escribe en la pantalla los números de línea según se va ejecutando el programa.
TROFF	Cancela TRON.
UPPER\$	Convierte minúsculas en mayúsculas.
VAL	Convierte un número que está en forma de cadena literal a forma numérica.
VPOS	Da la posición vertical del cursor de texto.
WEND	Señala el final de un bucle WHILE ... WEND.
WHILE	Crea un bucle que se repite mientras se cumpla la condición especificada.
WINDOW	Define una pantalla de texto.
XPOS	Da la posición horizontal del cursor gráfico.
YPOS	Da la posición vertical del cursor gráfico.
ZONE	Especifica la anchura de las zonas de escritura en la pantalla.



## Apéndice B

### Los parámetros de SOUND



#### SELECCION DE CANALES

Valor	Efecto
1	Seleccionar canal A
2	Seleccionar canal B
4	Seleccionar canal C
8	Sincronizar con A
16	Sincronizar con B
32	Sincronizar con C
64	Retener
128	Borrar cola

#### NUMEROS DE DURACION

Implícito: 20

Margen: -32768 a  
+32767

Valor	Efecto
> 0	Duración del sonido
= 0	Duración controlada por ENV
< 0	Número de veces que se repite la envolvente



## Apéndice C

---

### **Parámetros de la envolvente de volumen**

ENV n,P1,Q1,R1,P2,Q2,R2,P3,Q3,R3,P4,Q4,R4,P5,Q5,R5

n=número de la envolvente

Pn=número de escalones

Qn=amplitud de los escalones

Rn=duración de los escalones

Mínimo=una sección

Máximo=cinco secciones

Si se empieza a definir una sección, hay que incluir sus tres parámetros.

Si no se define ninguna sección (por ejemplo, ENV 1) la envolvente queda desactivada.





## Apéndice D

---

### **Parámetros de la envolvente de tono**

ENT n,T1,V1,W1,T2,V2,W2,T3,V3,W3,T4,V4,W4,T5,V5,W5

n=número de la envolvente

Tn=número de escalones

Vn=amplitud de los escalones

Wn=duración de los escalones

Mínimo=una sección

Máximo=cinco secciones

Si se empieza a definir una sección, hay que incluir sus tres parámetros.

Si no se define ninguna sección (por ejemplo, ENT 1) la envolvente queda desactivada.



## Apéndice E

---

### Tabla de notas, frecuencias y números de tono

DO media = C media = 1 (octava nº 0)

n	Nota	Frecuencia	Número de tono	Parámetro de octava (P)
-35	C	32.703	3822	-3
-34	C#	34.648	3608	-3
-33	D	36.708	3405	-3
-32	D#	38.891	3214	-3
-31	E	41.203	3034	-3
-30	F	43.654	2863	-3
-29	F#	46.249	2703	-3
-28	G	48.999	2551	-3
-27	G#	51.913	2408	-3
-26	A	55.000	2273	-3
-25	A#	58.270	2145	-3
-24	B	61.735	2025	-3
-23	C	65.406	1911	-2
-22	C#	69.296	1804	-2
-21	D	73.416	1703	-2
-20	D#	77.782	1607	-2
-19	E	82.407	1517	-2
-18	F	87.307	1432	-2
-17	F#	92.499	1351	-2
-16	G	97.999	1276	-2
-15	G#	103.826	1204	-2
-14	A	110.000	1136	-2

n	Nota	Frecuencia	Número de tono	Parámetro de octava (P)
-13	A#	116.541	1073	-2
-12	B	123.471	1012	-2
-11	C	130.813	956	-1
-10	C#	138.591	902	-1
-9	D	146.832	851	-1
-8	D#	155.564	804	-1
-7	E	164.814	758	-1
-6	F	174.614	716	-1
-5	F#	184.997	676	-1
-4	G	195.998	638	-1
-3	G#	207.652	602	-1
-2	A	220.000	568	-1
-1	A#	233.082	536	-1
0	B	246.942	506	-1
+1	C	261.626	478	0
+2	C#	277.183	451	0
+3	D	293.665	426	0
+4	D#	311.127	402	0
+5	E	329.628	379	0
+6	F	349.228	358	0
+7	F#	369.994	338	0
+8	G	391.995	319	0
+9	G#	415.305	301	0
+10	A	440.000	284	0
+11	A#	466.164	268	0
+12	B	493.883	253	0
+13	C	523.251	239	1
+14	C#	554.365	225	1
+15	D	587.330	213	1
+16	D#	622.254	201	1
+17	E	659.255	190	1
+18	F	698.457	179	1
+19	F#	739.989	169	1
+20	G	783.991	159	1
+21	G#	830.609	150	1
+22	A	880.000	142	1
+23	A#	932.328	134	1
+24	B	987.767	127	1
+25	C	1046.502	119	2
+26	C#	1108.731	113	2

n	Nota	Frecuencia	Número de tono	Parámetro de octava (P)
+27	D	1174.659	106	2
+28	D#	1244.508	100	2
+29	E	1318.510	95	2
+30	F	1396.913	89	2
+31	F#	1479.978	84	2
+32	G	1567.982	80	2
+33	G#	1661.219	75	2
+34	A	1760.000	71	2
+35	A#	1864.655	67	2
+36	B	1975.533	63	2
+37	C	2093.004	60	3
+38	C#	2217.461	56	3
+39	D	2349.318	53	3
+40	D#	2489.016	50	3
+41	E	2637.021	47	3
+42	F	2793.826	45	3
+43	F#	2959.955	42	3
+44	G	3135.963	40	3
+45	G#	3322.438	38	3
+46	A	3520.000	36	3
+47	A#	3729.310	34	3
+48	B	3951.066	32	3
+49	C	4186.009	30	4
+50	C#	4434.922	28	4
+51	D	4698.636	27	4
+52	D#	4978.032	25	4
+53	E	5274.041	24	4
+54	F	5587.652	22	4
+55	F#	5919.911	21	4
+56	G	6271.927	20	4
+57	G#	6644.875	19	4
+58	A	7040.000	18	4
+59	A#	7458.621	17	4
+60	B	7902.133	16	4



## Apéndice F

---

### **El generador de sonido: notas técnicas**

Estas notas van dirigidas a los lectores más interesados en detalles técnicos y a aquellos que quieran acceder al sistema operativo del ordenador Amstrad. Por supuesto, no dan una revisión completa del contenido de las ROM, sino solamente unas sugerencias para quienes deseen descifrar las complejidades del generador de sonidos.

El generador incluido en el Amstrad es el circuito integrado General Instruments AY-3-8912, que genera ondas cuadradas. El tono generado depende del período de la onda, el cual varía a incrementos de 8 microsegundos.

Las envolventes de volumen se pueden controlar por programa o directamente por los circuitos (por *hardware*). El control por programa ha sido el objeto de este libro. La alternativa es acceder directamente a los registros 11, 12 y 13 del generador. La selección de canales se realiza mediante los registros de control de amplitud (8 a 10). Si se va a utilizar una envolvente en el canal seleccionado, se debe poner a 1 el bit 4 del correspondiente registro de control de amplitud. Si, por el contrario, este bit está a 0, el volumen será controlado por los bits 0 a 3 del registro.

El registro 13 (bits 0 a 3) controla la forma de la envolvente. Por *hardware* se pueden controlar ocho envolventes. Sus características son las siguientes:

- 8: Ataque rápido y caída suave, repetidos.
- 9: Ataque rápido y caída suave, seguida de sostenimiento a amplitud cero.
- 10: Ataque rápido, seguido de caídas y ataques suaves repetidos.
- 11: Ataque rápido, caída suave, ataque rápido y sostenimiento a amplitud máxima.
- 12: Ataque suave y caída rápida, repetidos.

- 13: Ataque suave y sostenimiento a amplitud máxima.
- 14: Ataque y caída rápidos, repetidos.
- 15: Ataque suave, caída rápida y sostenimiento a amplitud cero.

El período de la envolvente determina la duración de las pendientes. El período es un número de 16 bits; los 8 bits menos significativos se guardan en el registro 11; los restantes, en el 12. Estos períodos son los intervalos de tiempo entre etapas de la envolvente; se miden en unidades de 128 microsegundos.

El registro 7 determina si se ha de incluir ruido en el sonido. Los bits 0 a 2 inhiben el tono en los canales A a C; los bits 3 a 5 inhiben el ruido en los canales A a C.

El generador de ruido produce un ruido pseudoaleatorio; su registro es el 6. Los registros 0 a 5 son los generadores de tono. Cada canal tiene dos registros de tono: uno es el de aproximación y el otro el de sintonía fina.

A continuación damos una lista de las direcciones de llamada de algunas rutinas del sistema operativo que pueden tener interés para el usuario.

- BCA7 Inicializa el generador de sonido y borra todas las colas.
- BCAA Envía un sonido a una cola.
- BCAD Comprueba si hay espacio libre en una cola de sonido.
- BCB0 Cuando la cola no está llena, activa una rutina.
- BCB3 Libera el sonido en los tres canales.
- BCB6 Retiene los sonidos.
- BCB9 Reanuda los sonidos retenidos por BCB6.
- BCBC Define una envolvente de volumen.
- BCBF Define una envolvente de tono.
- BCC2 Determina la posición de memoria en que se encuentran los datos de una envolvente de volumen.
- BCC5 Análoga a la anterior, pero referida a envolventes de tono.



¿Le gustaría convertirse en compositor,  
director de orquesta, solista y público con su  
Amstrad?

Entre, con la ayuda de Jeremy Vine, en el  
mundo musical y sonoro de este potente y  
versátil ordenador. En un estilo vivo y  
desenfadado, Jeremy le enseña a dominar las  
instrucciones de sonido y de envolventes a  
través de la experimentación práctica.  
¡Vecinos: alerta!

Para seguir este libro no es necesario que el  
lector tenga conocimientos de música, y muy  
pocos de BASIC. Vivirá una sonada aventura  
paseando por las páginas de este libro, y al  
mismo tiempo descubrirá:

- cómo componer y programar música
- cómo convertir su Amstrad en un  
sintetizador
- las interioridades del generador de sonido  
del Amstrad

y aprenderá a programar efectos sonoros de  
todo tipo:

- timbres de teléfono
- explosiones
- alarmas y sirenas
- escalas musicales
- campanas y silbidos

¿Le suena?

**AMSTRAD**  
**E S P A Ñ A**

Avda. del Mediterráneo, 9 28007 MADRID